z/OS

DFSMSdfp Storage Administration Reference

SC26-7402-01

z/OS

DFSMSdfp Storage Administration Reference

SC26-7402-01

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 421.

Second Edition, March 2002

This edition applies to Version 1 Release 3 of z/OS™ (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC26-7402-00.

Order publications through your IBM® representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

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

International Business Machines Corporation
RCF Processing, Department M86/050
5600 Cottle Road
San Jose, CA 95193-0001
United States of America

IBMLINK from US: STARPUBS at SJEVM5
IBMLINK from Canada: STARPUBS at TORIBM
IBM Mail Exchange: USIB3VVD at IBMMAIL
Internet: starpubs@us.ibm.com

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

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

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

© Copyright International Business Machines Corporation 1986, 2002. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures xi

Tables xv

About This Book xvii
Required Product Knowledge xvii
Referenced Publications xvii
Accessing z/OS DFSMS Books on the Internet xix
Using LookAt to look up message explanations xix
Accessing Licensed Books on the Web xix
How to Send Your Comments xx
Notational Conventions xxi

Summary of Changes xxiii
Summary of Changes for SC26-7402-01 z/OS DFSMSdfp xxiii
New Information xxiii
Deleted Information xxiv
Changed Information xxiv
Moved Information xxiv
Library Changes for z/OS DFSMS Version 1 Release 3 xxiv

Chapter 1. Introducing the Storage Management Subsystem 1
Understanding the Storage Management Subsystem 1
Using the Interactive Storage Management Facility 1
Defining your Storage Management Policy 3
Preparing for the Storage Management Subsystem 4
Running SMS in a Parallel Sysplex Environment 5
Basic Terms and Definitions 6
The Concept of System Grouping 6
Running SMS on Mixed DFSMS Releases 7

Chapter 2. Preparing for the Storage Management Subsystem 9
Allocating Control Data Sets 9
Calculating the Size of Storage and Active Control Data Sets 10
Calculating the Size of a COMMDS 12
Selecting Volumes for Control Data Sets 12
Allocating an SCDS 13
Allocating an ACDS 13
Allocating a COMMDS 14
Modifying the SYS1.PARMLIB Data Set 14
Initializing SMS Through the IGDSMSxx Member 14
Defining SMS to z/OS 23
Starting the SMS Address Space 25
Accessing the Storage Administrator Primary Option Menu 26
Planning for VSAM Record-Level Sharing 26

Chapter 3. Creating the Base Configuration 27
Planning the Base Configuration 27
Defining the Base Configuration 27
Specifying the Default Management Class 30
Specifying the Default Unit 31
Specifying the Default Device Geometry 31
Specifying the DS Separation Profile 31

© Copyright IBM Corp. 1986, 2002

Specifying Systems and System Groups in the SMS Complex 32
Defining the Base Configuration for VSAM Record-Level Sharing 33
Using Cache Structures 33
Defining Cache Sets 34
Completing the Base Configuration 35

Chapter 4. Defining Storage Groups 37
Understanding Storage Groups 37
DASD Storage Groups 38
DASD Volume Status for Data Sets 38
Object and Object Backup Storage Groups 39
OAM Collection Names 41
Tape Storage Groups 42
Planning Storage Groups for Data Sets 43
Planning Storage Groups for OAM Object Collections 44
Defining Storage Group Attributes 45
Defining a VIO Storage Group 46
Defining a Dummy Storage Group 49
Defining a Pool Storage Group 51
Assigning DASD Storage Groups to Data Sets 57
Defining an Object Storage Group 57
Defining the Object Backup Storage Groups 60
Defining Object or Object Backup Storage Group Status 62
Assigning an OAM Object Collection to a Storage Group 64
Defining a Tape Storage Group 64
Defining Tape Storage Group Status 65
Defining Additional Storage Groups 66
Listing Volumes in a Storage Group 66

Chapter 5. Defining Management Classes 69
Understanding Management Classes 69
Default Management Class 69
OAM Management Classes 70
Describing Management Classes 70
Planning Management Classes 71
Defining Management Class Attributes 71
Defining Management Class Expiration Attributes 73
Defining Management Class Migration Attributes 76
Defining Management Class Backup Attributes 79
Defining Object Class Transistion Attributes 81
Defining Aggregate Backup Attributes 83
Assigning Management Classes 84
Defining Additional Management Classes 84

Chapter 6. Defining Storage Classes 85
Understanding Storage Classes 85
Storage Classes for Data Sets 85
Storage Classes for Objects 86
Planning Storage Classes 87
Defining Storage Class Attributes 87
Defining Performance Objectives 88
Defining Availability 95
Defining Accessibility 95
Defining the Guaranteed Space Attribute 97
Defining Guaranteed Synchronous Write 99
Defining Use of the Coupling Facility for VSAM Record-Level Sharing . . . 99

Assigning Storage Classes 101
Defining Additional Storage Classes. 101
SMS Volume Selection for Data Set Allocation 102
Conventional Volume Selection 103
Striping Volume Selection 106
Possible Reasons for Volume Selection Failure 109

Chapter 7. Defining Data Classes 111
Understanding Data Classes 111
Planning Data Classes. 111
Defining Data Class Attributes 112
Defining Record and Space Attributes for Data Class 113
Defining Volume and Data Set Attributes for Data Class 115
Specifying Attributes to Handle Space Constraints During Allocation 119
Defining VSAM Attributes and Specifying Media Types for Data Class . . . 121
Defining Shareoptions and RLS Attributes for Data Class 123
Specifying Attributes for Data Set Reuse and Loading 125
Specifying Attributes for Backup-While-Open (BWO) and Recovery 125
Assigning Data Classes 127
Specifying Data Classes outside ACS Routines 127
Processing Data Class Attributes in JCL 127
Defining Additional Data Classes 128

Chapter 8. Defining Aggregate Groups 129
Understanding Aggregate Groups 129
Planning Aggregate Groups. 130
Defining Aggregate Groups 130
Defining Aggregate Group Attributes 131
Editing Aggregate Group Attributes 132
Backing up and Recovering an Aggregate Group 135
Defining Additional Aggregate Groups 135

Chapter 9. Defining ACS Routines 137
Understanding ACS Routines 137
Using ACS Routines for Data Sets Created by z/OS Network File System and Distributed FileManager/MVS 138
Restrictions on Using ACS Routines 138
Creating ACS Routines 138
Translating ACS Routines 140
Browsing the Results of a Translation 141
BROWSING the Results of an Unsuccessful Translation 142
Validating ACS Routines or an Entire SCDS. 143
Validating an ACS Routine 143
Validating an Entire SCDS 146
Translating and Validating in a Sysplex Environment 147
Testing ACS Routines 148
ACS Test Usage 148
Creating ACS Test Cases 148
Running ACS Test Cases 151
ACS Routines Invoked for Copying and Importing Data Sets 153
ACS Routines Invoked for Restoring, Recalling, Recovering, and Converting Data Sets 154
ACS Routine Environments 156
JCL DD Statement (Batch), and Dynamic Allocation 157
Volume Reference 157
Data Set Stacking 160

Non-Data Set Stacking Allocations 161
Tape Management System Support 163
Access Method Services 164
DFSMShsm 165
DFSMSdss 165
DFSMSrmm 166
ISMF 167
OAM 167
Processing of SMS Classes and Storage Groups 167
Displaying ACS Object Information 168
Deleting an ACS Object from an SCDS 169

Chapter 10. Activating Storage Management Subsystem Configurations 171
Manually Activating the First Storage Management Subsystem Configuration 171
Step One: IPL Each System in the SMS Complex 171
Step Two: Prepare One System 171
Step Three: Activate the Configuration from One System 171
Step Four: Activate SMS on the Other Systems 172
Automatically Activating a Storage Management Subsystem Configuration 172
Changing Storage Management Subsystem Parameters 173
Parameters of the SETSMS Operator Command 173
Considerations when Changing Storage Management Subsystem Configurations 177
OAM Considerations when Changing SCDSs 178

Chapter 11. Maintaining the Storage Management Subsystem. 179
Displaying SMS and OAM Information 179
Displaying Information about the Active Configuration 179
Displaying SMS TRACE Information 180
Displaying Storage Group Status Using the DISPLAY SMS Command 181
Displaying OAM Status in a Parallel Sysplex 183
Displaying the Caching Statistics 184
Displaying Storage Group Status Using ISMF 184
Displaying Volume Status Using ISMF 185
Displaying Volumes Using the DISPLAY SMS Command 187
Displaying the Device Status 188
Listing SMS Classes, Aggregate Groups, Storage Groups, and Libraries Using ISMF 195
Listing with View and Sort 195
Listing Data Classes 195
Listing Storage Groups, Management Classes, Storage Classes, Aggregate Groups, Libraries and Drives 200
Altering Data Set Associations 200
Altering SMS Classes, Aggregate Groups, Storage Groups, Libraries, Volumes, and Drives 200
Altering Storage Groups 200
Redefining Volumes Among Storage Groups 201
Altering Management Classes 202
Altering Storage Classes 203
Altering Data Classes 204
Altering Aggregate Groups 206
Altering the SCDS on Different DFSMS Releases. 209
Copying SMS Classes, Storage Groups, and Aggregate Groups 209
Deleting DASD Volumes from the System 210
Deleting DASD Volumes from Storage Groups 210
Deleting Storage Groups 211

Deleting or Moving System-Managed Data Sets from the Storage Group 211
Preventing Job Failures 211
Deleting SMS Classes. 212
Converting Volumes to SMS 213

Chapter 12. Recovering Storage Management Subsystem Information 215
Recovering Control Data Sets 215
Recovering an SCDS 215
Recovering an ACDS 215
Recovering a COMMDS 216
Recovering from a Systems Failure in the SMS Complex 217
Recovering the SMS Address Space 217
Canceling the SMS Address Space 217

Chapter 13. Protecting the Storage Management Subsystem 219
Identifying the Resource Owner and Extracting the Default Classes 219
Protecting ISMF Functions 220
Locating Module Names for ISMF Applications. 221
Locating Module Names for ISMF Functions 221
Locating Module Names for Line Operators and Commands. 222
Protecting Modules 227
Storage Administration (STGADMIN) Profiles in the FACILITY Class. 227
Command and Keyword Related Profiles 227
Authority to Activate a Storage Management Subsystem Configuration 231
Storage Administrator Facility Class Profiles for DFSMSdss 231
FIELD Resource Class 233
How Authorization and Protection of Classes Differ 233
Authorizing the Storage and Management Classes 233
Protecting the Control Data Sets 233
Authorizing for TSO and ISPF 233
Storage Administrator Authorization 234

Chapter 14. Administering VSAM Record-Level Sharing 235
Preparing for VSAM Record-Level Sharing 235
Determining Hardware Requirements 235
Understanding the Product Environment for VSAM RLS 236
Determining Applications That Can Use VSAM RLS 236
Ensuring Same Systems Connectivity 237
Planning for Availability 238
Defining Sharing Control Data Sets 239
Defining CF Cache Structures 241
Defining the CF Lock Structure 242
Modifying the SYS1.PARMLIB IGDSMSxx Member 247
Establishing Authorization for VSAM RLS. 247
Defining CF Cache Structures in the SMS Base Configuration 247
Defining Storage Classes for VSAM RLS. 250
Defining VSAM RLS Attributes in Data Classes 252
Activating VSAM RLS 252
Enabling VSAM RLS Processing 252
Enabling a Data Set for VSAM RLS Processing 253
Monitoring the Coupling Facility for VSAM RLS 253
Displaying CF Information 253
Changing the Size of Coupling Facility Cache Structures 255
Changing the Size of Coupling Facility Lock Structures. 256
Changing IGDSMSxx Parameters to Support the Coupling Facility 256
Changing the State of Coupling Facility Cache Structures and Volumes 258

Selecting Data Sets for Coupling Facility Statistical Monitoring 259
Recovering VSAM RLS Processing 259
Recovering the CF Lock Structure 259
Recovering a CF Cache Structure 260
Recovering the SMSVSAM Server Address Space 261
Recovering a Shared Control Data Set 261
Falling Back from VSAM RLS Processing 261
Fallback Rules and Considerations 262
Fallback Procedure 262

Chapter 15. Writing ACS Routines 265
Constants 265
Simple Mask Rules 266
Simple Mask Examples 266
Data Set Mask Rules 266
Data Set Mask Examples 267
Read-Write Variables 267
Using Read-Write Variables with Volume Reference (VOL=REF) 268
Using Read-Write Variables with Data Set Stacking 268
Read-Only Variables 269
Using Read-Only Variables 277
Constraints When Using Read-Only Variables 279
Comparison Operators 280
Comparison Rules 280
Boolean Expressions 280
Special Functions 281
Determining Distributed FileManager/MVS Data Set Creation Requests 282
Statements 284
PROC Statement 285
FILTLIST Statement 285
SET Statement 286
DO Statement 287
IF Statement 288
SELECT Statement 289
EXIT Statement 290
WRITE Statement 291
END Statement 292
Sample ACS Routine 292

Appendix A. Quick Reference to ISMF Commands and Line Operators 295

Appendix B. SETCACHE Functions and Device Information 303
Maintaining the Media Characteristics of a Volume 303
Modifying DASD Storage Control Characteristics 303
Modifying Caching Characteristics 304
Modifying Duplexing Characteristics 305
Modifying Destaging Characteristics 306
Submitting Jobs 306
Providing Remote Authorization Codes 306

Appendix C. Space Utilization and Capacity Planning 309
Using the Data Collection Application 309

Appendix D. Using Data Set Separation 315
Overview 315
Syntax for Creating a Data Set Separation Profile 316

Steps for Creating a Data Set Separation Profile 317
Creating Multiple Data Set Separation Profiles 318
Using Data Set Separation with Generation Data Groups and Striping 318
Environmental Conditions Affecting Data Set Separation 319
Factors Affecting Code Path Length 319

Appendix E. Using NaviQuest 321
How to Start NaviQuest 321
Terminology 322
Methods for Collecting Test Data 324
ISMF Lists 324
DCOLLECT Data 324
SMF Data 325
VMA Data 325
Data Classification Methodology 325
Testing ACS Routines 326
Setting Up the Test Environment 326
Testing Procedures 327
NaviQuest Testing Scenario 339
Performing Storage Administration Tasks in Batch 343
How to Run Storage Administration Tasks in Batch 343
Sample JCL for Batch 344
NaviQuest CLISTS and REXX EXECS 349
Storage Group Volume Add/Delete: ACBQBAI9 369
ACS Routine Translate: ACBQBAO1 372
SCDS Validation: ACBQBAO2 376
Test ACS Routines: ACBQBAIA 378
Generate Test Cases from ISMF-saved Data Set Lists: ACBQBAG3 380
Generate Test Cases from DCOLLECT Data: ACBQBAG1 381
Generate Test Cases from SMF Data 382
Generate Test Cases from VMA Extract Data: ACBQBAO3 383
ACS Test Listings Comparison: ACBQBAC1 384
Enhanced ACS Test Listing: ACBQBAX1 386
Update Test Cases with Expected Results: ACBQBAU1 388
Generate Report from ISMF-saved Data Set List: ACBQBAR1 389
Generate Report from ISMF-saved DASD Volume List: ACBQVAR1 393
Generate Report from ISMF-Saved Tape List: ACBQBAR4 396
Generate Data Set Report from DCOLLECT Data: ACBQBAR7 399
Generate Model Commands from ISMF-saved List: ACBQBAM1 402
Creating DFSMS Online Reports 402
Creating Data Set Reports from Saved ISMF Lists 403
Creating Volume Reports from Saved ISMF Lists 405
Creating Customized Tape Reports from Saved ISMF Lists 406
Data Set Report from DCOLLECT Data 407
DASD Volume Report from DCOLLECT Data 408
SMS Configuration Report from DCOLLECT Data 409
Additional Storage Administration Functions 410
QSAVE and QRETRIEV Commands: Saving and Retrieving ISMF Selection Criteria 410
Creating Model Commands 412
COPYFILT Macro: COPYLIB Facility for FILTLISTs 416

Appendix F. Accessibility 419
Using assistive technologies 419
Keyboard navigation of the user interface 419

Notices 421
Programming Interface Information 422
Trademarks. 422

Glossary 423

Index 437

Figures

1. ISMF Primary Option Menu for Storage Administrators . . . 2
2. Activating an SMS Configuration . . . 4
3. System Grouping Concept . . . 7
4. Relationship among SCDSs and ACDSs in an Installation . . . 10
5. CDS Application Selection . . . 28
6. SCDS Base Define panel . . . 29
7. SCDS Base Define panel . . . 30
8. CDS Application Selection for VSAM RLS . . . 34
9. Cache Set Update Panel for VSAM RLS . . . 35
10. Formula for Deriving Optimum Number of Stripes . . . 43
11. Defining a Storage Group . . . 45
12. Defining VIO Storage Group Attributes . . . 47
13. Defining Dummy Storage Group Attributes . . . 49
14. Defining Dummy Storage Group Volume Serial Numbers . . . 50
15. Defining Pool Storage Group Attributes . . . 51
16. Defining Pool Storage Group System Status . . . 55
17. Defining Volume System Status . . . 56
18. Defining Object Storage Group Attributes . . . 58
19. Defining Object Backup Storage Group Attributes . . . 61
20. Defining Object or Object Backup Storage Group System Status . . . 63
21. Defining Tape Storage Group Attributes . . . 64
22. Defining Tape Storage Group System Status . . . 65
23. Management Class Application Selection Panel for Storage Administrators, Page 1 of 2 . . . 72
24. Management Class Application Selection Panel for Storage Administrators, Page 2 of 2 . . . 73
25. Defining Management Class Expiration Attributes . . . 74
26. Defining Management Class Migration Attributes . . . 76
27. Defining Management Class Backup Attributes . . . 79
28. Defining Object Class Transition Attributes . . . 82
29. Defining Aggregate Backup Attributes . . . 83
30. Defining a Storage Class . . . 87
31. Defining Storage Class Attributes, Page 1 of 2 . . . 88
32. MSR Capabilities . . . 90
33. Allocation of Primary and Secondary Space for Multivolume Data Sets . . . 98
34. Defining Storage Class Attributes, Page 2 of 2 . . . 100
35. Example of Evaluation Process for Volume Selection . . . 104
36. Defining a Data Class . . . 112
37. Defining Data Class Record and Space Attributes . . . 113
38. Defining Data Class Volume and Data Set Attributes . . . 116
39. Defining Data Class VSAM Attributes and Media Types . . . 121
40. Defining Data Class Shareoptions and RLS Attributes . . . 124
41. Aggregate Group Application Selection . . . 131
42. Aggregate Group Define, Page 1 of 2 . . . 132
43. Aggregate Group Define, Page 2 of 2 . . . 133
44. Editing an Aggregate Group Selection Data Set- Example . . . 134
45. Writing an ACS Routine . . . 139
46. Invoking the ISPF/PDF Editor . . . 139
47. Translating an ACS Routine . . . 140
48. Browsing the Results of an ACS Translation . . . 141
49. Generating an ACS Translation Listing . . . 142
50. Browsing the Results of an Unsuccessful ACS Translation . . . 143
51. Validating an ACS Routine . . . 144
52. Browsing the Results of an ACS Routine Validation . . . 145
53. Validating an SCDS . . . 146

© Copyright IBM Corp. 1986, 2002

54. ACS Test Case Selection panel . . . 149
55. Creating ACS Test Cases, Page 1 of 4 . . . 149
56. Creating ACS Test Cases, Page 2 of 4 . . . 150
57. Creating ACS Test Cases, Page 3 of 4 . . . 150
58. Creating ACS Test Cases, Page 4 of 4 . . . 151
59. Testing an ACS Routine . . . 152
60. Results of Testing an ACS Routine . . . 152
61. Example of REF=ST Values when Using VOL=REF . . . 158
62. Example of REF=NS Values when Using VOL=REF . . . 158
63. Example of Failing VOL=REF Values That Are Not Valid . . . 159
64. Example of a Storage Class ACS Routine for Read-Only Variables . . . 161
65. Example of a Storage Group ACS Routine for Read-Only Variables . . . 161
66. Example of a Storage Class ACS Routine Specifying Both Volume Reference and Unit Affinity 163
67. Displaying ACS Object Information . . . 168
68. Deleting ACS Objects . . . 169
69. Displaying Information about the Active Configuration . . . 180
70. Displaying Trace Information . . . 181
71. DISPLAY SMS,STORGRP Command Syntax . . . 181
72. Displaying Storage Group Status Information . . . 182
73. Displaying Storage Group Volume Status Information . . . 183
74. Displaying Cache and DASD Fast Write Information . . . 184
75. Displaying Storage Group Status for Each System/System Group . . . 185
76. Displaying the SMS Volume Status . . . 186
77. Displaying Status Information for Individual DASD Volumes . . . 187
78. DEVSERV S,430 Output . . . 188
79. DEVSERV SMS Command Syntax . . . 189
80. DEVSERV QPAVS Display Content . . . 190
81. DS QP,D123,3 Output . . . 190
82. DS QP,D2FF,VOLUME Output . . . 191
83. DS QP,D222,VOLUME Output . . . 191
84. DS QP,D6FF,UNBOX Output . . . 191
85. DS QP,D5EF,UCB Output . . . 192
86. DS QP,SSID=1401 Output . . . 192
87. DS QP,D345 Output . . . 193
88. DS QP,D621 Output . . . 193
89. DS QP,D6F4 Output . . . 193
90. DS QP,F60 Output . . . 194
91. DS P,A000 Output . . . 194
92. DS QD,D407,RDC Output . . . 194
93. Specifying Data Class View Criteria, Page 1 of 3 . . . 196
94. Specifying Data Class View Criteria, Page 2 of 3 . . . 197
95. Specifying Data Class View Criteria, Page 3 of 3 . . . 197
96. Specifying Data Class Sort Criteria, Page 1 of 3 . . . 198
97. Specifying Data Class Sort Criteria, Page 2 of 3 . . . 198
98. Specifying Data Class Sort Criteria, Page 3 of 3 . . . 199
99. Displaying a List of Data Classes . . . 199
100. Moving a Volume from One Storage Group to Another . . . 201
101. Altering a Management Class . . . 202
102. Altering a Storage Class . . . 203
103. Data Class Alter Panel, Page 1 of 4 . . . 204
104. Data Class Alter Panel, Page 2 of 4 . . . 205
105. Data Class Alter Panel, Page 3 of 4 . . . 205
106. Data Class Alter Panel, Page 4 of 4 . . . 206
107. Aggregate Group Alter, Page 1 of 2 . . . 207
108. Aggregate Group Alter, Page 2 of 2 . . . 208
109. Copy Entry Panel . . . 209

110. Confirm Delete Request Panel . . . 213
111. Example of Global Connectivity in a Parallel Sysplex . . . 237
112. Example of Insufficient Connectivity in a Parallel Sysplex . . . 238
113. JCL Example for Creating Sharing Control Data Sets . . . 240
114. CDS (Control Data Set) Application Selection for VSAM RLS . . . 248
115. Coupling Facility Cache Set Definition for VSAM RLS . . . 249
116. Example of CF Cache Structure Definition in Base Configuration . . . 250
117. Defining a Storage Class for VSAM RLS . . . 251
118. Defining Storage Class Attributes for VSAM RLS . . . 251
119. Example of Constraints when Using the &ALLVOL and &ANYVOL Read-Only Variables . . . 282
120. Example Showing &JOB Read-Only Variable . . . 283
121. Example Showing &PGM Read-Only Variable . . . 283
122. Example Showing &USER Read-Only Variable . . . 284
123. Example of a FILTLIST Statement . . . 286
124. Using the INCLUDE and EXCLUDE . . . 286
125. Example of a DO Statement . . . 288
126. Examples of IF Statements . . . 288
127. Example of a SELECT Statement . . . 290
128. Example of an EXIT Statement . . . 290
129. Example of an END Statement . . . 292
130. Production ACS Routine for Storage Class . . . 292
131. Entry Panel for the Data Collection Application, Page 1 of 3 . . . 310
132. Entry Panel for the Data Collection Application, Page 2 of 3 . . . 312
133. Entry Panel for the Data Collection Application, Page 3 of 3 . . . 313
134. ISMF Primary Option Menu for Storage Administrators, DGTSMMD2 . . . 322
135. NaviQuest Primary Option Menu, ACBSMDP0 . . . 322
136. Typical Data Types . . . 325
137. TSO Data Type with Data Subtypes . . . 326
138. Test Case Generation Selection Menu, ACBSFLG4 . . . 329
139. Test Case Generation from Saved ISMF List Entry Panel, ACBDFLG1 . . . 330
140. Test Case Generation from DCOLLECT Data Entry Panel, ACBDFLG5 . . . 330
141. Test Case Generation from SMF Data Entry Panel, ACBDFLG3 . . . 331
142. Test Case Generation from VMA Extract Data Entry Panel, ACBDFLG2 . . . 332
143. NaviQuest Primary Option Menu, ACBSMDP0 . . . 334
144. ACS Test Listings Comparison Entry Panel, ACBDFLC1 . . . 335
145. Enhanced ACS Test Listing Entry Panel, ACBDFLX1 . . . 337
146. Test Case Update with Test Results Entry Panel, ACBDFLU1 . . . 338
147. Testing Example Flowchart . . . 341
148. Batch Testing/Configuration Management Selection Menu, ACBSMDJ1 . . . 343
149. SYS1.SACBCNTL member ACBJBAOB JCL for Batch Functions . . . 347
150. Sample JCL for ACBJBAI2 . . . 354
151. Sample JCL for ACBJBAI4 . . . 358
152. Sample JCL for ACBJBAIA . . . 361
153. Sample JCL for ACBJBAJ1 . . . 363
154. Sample JCL for ACBJBAIB . . . 370
155. Sample JCL for ACBJBAC2 . . . 372
156. Sample JCL for ACBJBAOS . . . 377
157. Sample JCL for ACBJBAIC . . . 379
158. Sample JCL for ACBJBAG2 . . . 380
159. Sample JCL for ACBJBAG1 . . . 381
160. Sample JCL for ACBJBAI1 . . . 382
161. Sample JCL for ACBJBAOW . . . 383
162. Sample JCL for ACBJBAC1 . . . 384
163. Sample JCL for ACBJBAX1 . . . 386
164. Sample JCL for ACBJBAU2 . . . 388
165. Sample JCL for ACBJBAOD . . . 391

166. Sample JCL for ACBJBAOF 394
167. Sample JCL for ACBJBAOT 397
168. Sample JCL for ACBJBARD 400
169. Sample JCL for ACBJBAM1 402
170. NaviQuest Primary Option Menu, ACBSMDP0 403
171. SMS Report Generation Selection Menu, ACBSFLR1 404
172. Data Set Report from Saved ISMF List Entry Panel, ACBDRPR5 405
173. DASD Volume Report from Saved ISMF List Entry Panel, ACBDRPR7 406
174. Tape Volume Report from Saved ISMF List Entry Panel, ACBDFLR3 407
175. Data Set Report from DCOLLECT Data Entry Panel, ACBDRPR1 408
176. DASD Volume Report from DCOLLECT Data Entry Panel, ACBDRPR2 409
177. SMS Configuration Report from DCOLLECT Data Entry Panel, ACBDRPR3 410
178. Data Set Selection Entry Panel, DGTDDDS1 412
179. Model Commands Generation Selection Menu, ACBSUTM3 413
180. Model Commands Generation from Saved ISMF List Entry Panel, ACBDUTM1 414
181. Model Commands Generation from DCOLLECT Data Entry Panel, ACBDUTM2 414
182. Copy FILTLISTs Entry Panel, ACBDFIO1 417

Tables

1. Parameters of the IGDSMS*xx* Member of SYS1.PARMLIB . . . 15
2. Volumes Listed in a Specified Storage Group . . . 66
3. Comparing Retention Period Attributes . . . 75
4. D/T3990 SMS Cache Candidate Tokens for Sequential and Direct Requests . . . 92
5. Combinations for Requesting Point-in-Time Copy Devices . . . 97
6. Volume Selection Preferences . . . 103
7. Applying Data Class Attributes to Record Organization (Recorg) . . . 122
8. Results of Output Listing Disposition . . . 142
9. Allocation, IMPORT, and COPY Conditions . . . 153
10. RESTORE, RECALL, RECOVER, CONVERTV, and FORCENONSMS Conditions . . . 155
11. Values for &UNIT ACS Read-Only Variable . . . 161
12. AFFing DD Volser Values . . . 162
13. DFSMSrmm Read-Only Variables . . . 166
14. Module Names for ISMF Applications . . . 221
15. Module Names for ISMF Functions . . . 221
16. Module/CLIST Names for ISMF Line Operators, Part I . . . 223
17. Module/CLIST Names for ISMF Line Operators, Part 2 . . . 224
18. Module/CLIST Names for ISMF Line Operators, Part 3 . . . 225
19. Module/CLIST Names for ISMF Commands . . . 226
20. Effect of MAXSYSTEM Value on Lock Table Entry Size . . . 243
21. CF Lock Structure Sizing Examples . . . 243
22. Using Read-Write Variables in ACS Routines . . . 268
23. Read-Only Variables . . . 269
24. Indexing Functions for Read-Only Variables . . . 281
25. Read-Write Variable Assignments . . . 287
26. ISMF Commands . . . 295
27. ISMF Line Operators . . . 298
28. SYS1.SACBCNTL Sample JCL Library Member List . . . 345
29. CLISTs and REXX EXECs for Storage Administration Tasks . . . 349
30. Using ACBQBA12 Parameters . . . 351
31. ACBQBAI4 Parameters . . . 356
32. ACBQBAID Parameters . . . 360

© Copyright IBM Corp. 1986, 2002

About This Book

This book introduces system programmers and storage administrators to the IBM™ Storage Management Subsystem (SMS) and storage management concepts. More importantly, it describes in detail how to define, initialize, and maintain SMS and how to manage storage with the IBM Interactive Storage Management Facility (ISMF).

This book is intended for system programmers and storage administrators like you who manage storage under DFSMS™. As its title indicates, this section summarizes specific changes to the current release. If you are new to SMS and ISMF, you should start with “Chapter 1. Introducing the Storage Management Subsystem” on page 1 so that you can familiarize yourself with the products first. You may also refer to the publications listed in ″Referenced Publications″ on page xvii for further information on storage management concepts and ISMF applications.

Required Product Knowledge

To use this book effectively, you should be familiar with:

- Storage management concepts
- ISMF applications
- DFSMS Data Set Services (DFSMSdss™)
- DFSMS Hierarchical Storage Manager (DFSMShsm™)
- Device Support Facility (ICKDSF)
- Object Access Method (OAM)

Referenced Publications

The following publications are referenced in this book:

Publication Title	Order Number
Character Data Representation Architecture Reference and Registry	SC09-2190
CICS Recovery and Restart Guide	SC34-5721
CICS Transaction Server for OS/390 Migration Guide	GC34-5699
CICS Transaction Server for OS/390: Planning for Installation	GC34-5700
CICS Transaction Server for OS/390 Release Guide	GC34-5701
Device Support Facilities User’s Guide and Reference	GC35-0033
IBM 3495 Tape Library Dataserver Introduction	GA32-0234
IBM 3495 Tape Library Dataserver Operator’s Guide	GA32-0235
IBM Enterprise Storage Server Introduction and Planning Guide	GC26-7294
IBM Enterprise Storage Server User’s Guide	SC26-7295
International Technical Support Centers Implementing Concurrent Copy	GG24-3990
MVS/ESA™ System Messages, Volume 1	GC28-1480
MVS/ESA System Messages, Volume 2	GC28-1481
MVS/ESA System Messages, Volume 3	GC28-1482
MVS/ESA System Messages, Volume 4	GC28-1483
MVS/ESA System Messages, Volume 5	GC28-1484

© Copyright IBM Corp. 1986, 2002

Publication Title	Order Number
MVS/ESA SML: Managing Data	SC26-3124
MVS/ESA SML: Managing Storage Groups	SC26-3125
z/OS DFSMS Access Method Services	SC26-7394
z/OS DFSMS Advanced Copy Services	SC35-0428
z/OS DFSMS: Implementing System-Managed Storage	SC26-7407
z/OS DFSMS Installation Exits	SC26-7396
z/OS DFSMS Introduction	SC26-7397
z/OS DFSMS: Managing Catalogs	SC26-7409
z/OS DFSMS Migration	GC26-7398
z/OS DFSMS OAM Application Programmer's Reference	SC35-0425
z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support	SC35-0426
z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	SC35-0427
z/OS DFSMS Software Support for the IBM Enterprise Storage Server	SC26-7399
z/OS DFSMS: Using Data Sets	SC26-7410
z/OS DFSMS: Using the Interactive Storage Management Facility	SC26-7411
z/OS DFSMSdfp Diagnosis Reference	GY27-7618
z/OS DFSMSdss Storage Administration Guide	SC35-0423
z/OS DFSMSdss Storage Administration Reference	SC35-0424
z/OS DFSMShsm Data Recovery Scenarios	GC35-0419
z/OS DFSMShsm Storage Administration Guide	SC35-0421
z/OS DFSMSrmm Implementation and Customization Guide	SC26-7405
z/OS ISPF User's Guide Volume I	SC34-4822
z/OS MVS Initialization and Tuning Guide	SA22-7591
z/OS MVS JCL Reference	SA22-7597
z/OS MVS JCL User's Guide	SA22-7598
z/OS MVS Setting Up a Sysplex	GC28-1779
z/OS MVS Programming: Sysplex Services Guide	SA22-7617
z/OS MVS System Commands	SA22-7627
z/OS MVS System Management Facilities (SMF)	SA22-7630
z/OS Parallel Sysplex Application Migration	SA22-7662
z/OS Security Server RACF Security Administrator's Guide	SA22-7683
z/OS TSO/E Customization	SA22-7783
Storage Subsystem Library Master Bibliography, Index, and Glossary	GC26-4496

Accessing z/OS DFSMS Books on the Internet

In addition to making softcopy books available on CD-ROM, IBM provides access to unlicensed z/OS softcopy books on the Internet. To find z/OS books on the Internet, first go to the z/OS home page: http://www.ibm.com/servers/eserver/zseries/zos

From this Web site, you can link directly to the z/OS softcopy books by selecting the Library icon. You can also link to IBM Direct to order hardcopy books.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages, system abends, and some codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html`

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS).

To find a message explanation on the Internet, go to the LookAt Web site and simply enter the message identifier (for example, `IAT1836` or `IAT*`). You can select a specific release to narrow your search. You can also download code from the *z/OS Collection*, SK3T-4269 and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt as a TSO command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO from a disk on your *z/OS Collection*, SK3T-4269 or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html.
2. Click the **News** button.
3. Scroll to **Download LookAt Code for TSO and VM.**
4. Click the ftp link, which will take you to a list of operating systems. Select the appropriate operating system. Then select the appropriate release.
5. Find the **lookat.me** file and follow its detailed instructions.

To find a message explanation from a TSO command line, simply enter: **lookat** *message-id*. LookAt will display the message explanation for the message requested.

Note: Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt prompts you to choose which book to open.

Accessing Licensed Books on the Web

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

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

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password log on to:
`http://www.ibm.com/servers/resourcelink`

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

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile.**
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed, you will receive an e-mail confirmation.

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

To access the licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DFSMS documentation:

- Send your comments by e-mail to:
  - IBMLink™ from US: starpubs@us.ibm.com
  - IBMLink from Canada: STARPUBS at TORIBM
  - IBM Mail Exchange: USIB3VVD at IBMMAIL
  - Internet: starpubs@us.ibm.com

  Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).
- Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to IBM Corporation, RCF Processing Department M86/050, 5600 Cottle Road, San Jose, California 95193-0001, U.S.A.

Notational Conventions

A uniform notation describes the syntax of commands. This notation is not part of the language; it is merely a way of describing the syntax of the commands. The command syntax definitions in this book use the following conventions:

[ ] Brackets enclose an optional entry. You can, but need not, include the entry. Examples are:
[*length***]**
[MF=E]

| An OR sign (a vertical bar) separates alternative entries. You must specify one, and only one, of the entries unless you allow an indicated default. Examples are:
[REREAD|LEAVE]
[*length***|'S']**

{ } Braces enclose alternative entries. You must use one, and only one, of the entries. Examples are:
BFTEK={S|A}
{K|D}
{*address***|S|O}**

Sometimes alternative entries are shown in a vertical stack of braces. An example is:

```
MACRF={{(R[C|P])}{(W[C|P|L])}
       {(R[C],W[C])}}
```

In the example above, you must choose only one entry from the vertical stack.

. . . An ellipsis indicates that the entry immediately preceding the ellipsis might be repeated. For example:
(*dcbaddr***,[(***options***)],. . .)**

‘ ’ A **‘ ’** indicates that a blank (an empty space) must be present before the next parameter.

UPPERCASE BOLDFACE
Uppercase boldface type indicates entries that you must code exactly as shown. These entries consist of keywords and the following punctuation symbols: commas, parentheses, and equal signs. Examples are:
CLOSE , , , ,TYPE=T
MACRF=(PL,PTC)

UNDERSCORED UPPERCASE BOLDFACE
Underscored uppercase boldface type indicates the default used if you do not specify any of the alternatives. Examples are:
[EROPT={ACC|SKP|ABE}]
[BFALN={F|D}]

Lowercase Italic
Lowercase italic type indicates a value to be supplied by you, the user, usually according to specifications and limits described for each parameter. Examples are:
number
image-id
count.

Summary of Changes

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

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

Summary of Changes for SC26-7402-01 z/OS DFSMSdfp

This book contains information previously presented in *OS/390™ DFSMSdfp Storage Administration Reference*, SC35-7402-00.

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- DFSMS supports greater than 4K coupling facility caching for VSAM spheres that are opened for RLS processing. The new SMS data class keyword RLS CF Cache Value and its values (NONE, UPDATESONLY and ALL) are used to specify the level of caching requested. The new SMS parmlib keyword RLS_MaxCfFeature level controls the availability of the new function. The SET SMS=xx and the SETSMS commands have been modified to allow the RLS_MaxCfFeatureLevel keyword. The D SMS,OPTION command and the MVS commands D SMS,SMSVSAM, D SMS,SMSVSAM,ALL and D SMS,CFCACHE have also been modified.
- DFSMS supports dynamic volume count, which enables dynamic extend of volume count to SMS data sets. The data class parameter Dynamic Volume Count is used to determine the number of volumes to dynamically add to a data set when a new volume is needed.
- DFSMS supports data set separation, which automates separation of mission-critical data to reduce the impact of single points of failure. To utilize data set separation, you create a data set separation profile and specify it in the SMS base configuration. During volume selection for data set allocation, SMS attempts to separate, on the PCU level, the data sets that are listed in the profile.
- DFSMS supports the use of designated extend storage groups, each of which is associated with a primary storage group. This allows a data set to be extended when the primary storage group is out of space, so that processing can continue.
- DFSMS supports the use of designated overflow storage groups, which serve as reserve areas for primary space allocations during periods of high space utilization. In volume selection, volumes in overflow storage groups are preferred over quiesced volumes.
- OAM supports multiple object backup, which allows a system within an SMS complex to have more than one object backup storage group. Up to two object backup storage groups may be associated with each object storage group. Separate backup copies of objects, that are physically based on the object storage group to which the object belongs, may be specified.

© Copyright IBM Corp. 1986, 2002

- DFSMS supports system-managed duplexing rebuild for coupling facility (CF) lock structures, which improves failure recovery capability by creating a duplex copy of a structure in advance of any failure. If a failure occurs, the system automatically begins to use the duplexed copy, reducing loss of data. In addition, the steps associated with system-managed duplexing rebuild are much simpler than those associated with user-managed duplexing rebuild.
- NaviQuest supports the deletion of configurations through batch.

Deleted Information

The following information was deleted from this edition:

- Removed data class attribute fields *Imbed* and *Replicate* and their associated term definitions from the Data Class Define panel.
- Removed information about optical and tape status displays from “Displaying OAM Status in a Parallel Sysplex” on page 183, and referred users to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.
- Removed information about optical and tape status displays from “Displaying Volumes Using the DISPLAY SMS Command” on page 187, and referred users to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Changed Information

The following information was changed in this edition:

- Changed definition of data class attribute field *Volume Count* to increase accuracy and clarity.

Moved Information

- The NaviQuest™ information previously available in *DFSMS/MVS V1R5 NaviQuest User’s Guide* has been moved to “Appendix E. Using NaviQuest” on page 321.

Library Changes for z/OS DFSMS Version 1 Release 3

You might notice changes in the style and structure of some content in this book—for example, headings that are more task-oriented, notes with headings that are more specific and clear in their intent, additional index entries for easier information retrieval, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

Chapter 1. Introducing the Storage Management Subsystem

This chapter introduces the Storage Management Subsystem (SMS). In doing so, it explains various applications of the Interactive Storage Management Facility (ISMF) so that you can use them to define your storage management policy and manage your SMS configurations. In addition, it provides an overview of the preparatory tasks for implementing SMS and explores considerations for running SMS in a Parallel Sysplex™ environment and on mixed DFSMS™ releases.

Understanding the Storage Management Subsystem

The Storage Management Subsystem (SMS) is a DFSMS facility designed for automating and centralizing storage management. Using SMS, you can describe data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the systems.

SMS improves storage space use, allows central control of external storage, and enables you to manage storage growth more efficiently. With SMS, you can easily manage conversion between device types and ultimately move toward system-managed storage.

Using the Interactive Storage Management Facility

The Interactive Storage Management Facility (ISMF) provides a series of applications for storage administrators to define and manage SMS configurations. You can use these applications to:

- Define SMS base configuration information.
- Define, alter, delete, or copy individual SMS classes, storage groups, aggregate groups, optical libraries, optical drives, and tape libraries.
- Display parameters and values of individual SMS classes, storage groups, aggregate groups, mountable optical volumes, optical drives, mountable tape volumes, tape libraries and optical libraries.
- Generate, save and manage lists of SMS classes, storage groups, aggregate groups, mountable optical volumes, optical libraries, optical drives, mountable tape volumes and tape libraries.
- Edit ACS routines.
- Define, alter, and execute ACS test cases.
- Validate the correctness and completeness of an SMS configuration.
- Activate an SMS configuration.
- Display, define, alter, or delete storage group information pertaining to specific volumes using AUDIT, EJECT, ALTER, and RECOVER (RECOVER is for optical volumes only).
- Produce data set, volume, or capacity planning measurement data.
- Maintain mountable optical volumes and mountable tape volumes.
- Use DFSMSrmm™ to maintain tape volumes.
- Use the DFSMS NaviQuest tool to perform enhanced testing of your ACS routines, and to perform many storage management tasks in batch, such as:
  - Updating and testing your base configuration
  - Translating and testing your ACS routines
  - Generating test cases from previously collected DCOLLECT data

© Copyright IBM Corp. 1986, 2002

– Defining, altering, and displaying information for management classes, data classes and storage classes
– Defining or altering information for storage groups
– Defining, altering, and displaying information for the base configuration, as well as for aggregate groups
– Generating data set and volume lists and reports
– Diagnosing data set and volume problems

See “Appendix E. Using NaviQuest” on page 321 for more information on the DFSMS NaviQuest tool.

Figure 1 shows the ISMF Primary Option Menu for storage administrators.

```
  Panel  Help
------------------------------------------------------------------------------
                    ISMF PRIMARY OPTION MENU - DFSMS V1 R3
Enter Selection or Command ===>

Select one of the following options and press Enter:

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

Figure 1. ISMF Primary Option Menu for Storage Administrators

This primary option menu differs from the one that end users see. To the options found on the ISMF Primary Option Menu for End Users, this primary option menu adds the following:
- Storage Group
- Automatic Class Selection
- Control Data Set
- Library Management
- Data Collection

Additionally, the management class, data class, storage class, and aggregate group applications available through the ISMF Primary Option Menu for Storage Administrators allow you to define, alter, copy, and delete SMS classes. End users can list the available SMS classes, display the attributes of individual SMS classes, and list or display volumes and data sets.

To learn how to use the ISMF Data Collection application, see “Appendix C. Space Utilization and Capacity Planning” on page 309.

Defining your Storage Management Policy

SMS manages an installation's storage according to the currently active storage management policy. Through ISMF, you define an installation storage management policy in an *SMS configuration*. An SMS configuration contains the following:

- Base configuration information
- Classes and groups
- Automatic class selection (ACS) routines
- Optical library and drive definitions
- Tape library definitions

The *base configuration* identifies the systems that the SMS configuration manages. These systems constitute an *SMS complex*. The base configuration also contains installation defaults.

You can define more than one control data set, but only one at a time controls SMS. Each control data set defined for SMS is called a *source control data set (SCDS)*. The control data set that is in effect at a given time is the *active control data set (ACDS).*

SMS classes and groups are lists of traits and characteristics that are associated with or assigned to data sets, objects and volumes. An SMS configuration can contain the following five types of classes and groups:

Storage group allows you to define a list of volumes and manage them as if they were one large, single volume. SMS applies the properties you assign to a storage group to all the volumes within the storage group.

Management class allows you to define different levels of migration, backup and retention services. Through management class, you can associate a level of service with a data set or object that is independent of the physical location of the data set or object. Also, you can identify an object characteristic that might trigger a class transition.

Storage class allows you to define different levels of performance and availability services. Through storage class, you can separate the level of service for a data set or object from physical device characteristics. You can also separate the level of service for an object with different storage classes used to place objects at various levels of the storage hierarchy.

Data class allows you to define allocation defaults. Through data class, you can simplify and standardize the allocation of new data sets.

Aggregate Group allows you to define groups of data sets for the purpose of backing up or recovering all data sets in a group in a single operation.

An SMS configuration can contain multiple constructs of each type. Data sets managed by SMS are called *system-managed*. Each system-managed data set or object must reside in a storage group. The system-managed data sets must have a storage class, and might also have a management class and a data class. The objects must have a storage class and a management class.

You can assign the same name to various SMS classes and a storage group. For example, a data class and a storage class can have the same name.

ACS routines determine the SMS classes and storage groups for data sets and objects. You can also use ACS routines to control the transition of data sets to and from SMS management (objects are always SMS-managed).

Preparing for the Storage Management Subsystem

Before you implement a storage management policy by activating an SMS configuration, you need to follow the steps outlined in Figure 2. The chapters in this manual that describe these steps appear to the right of the respective boxes.

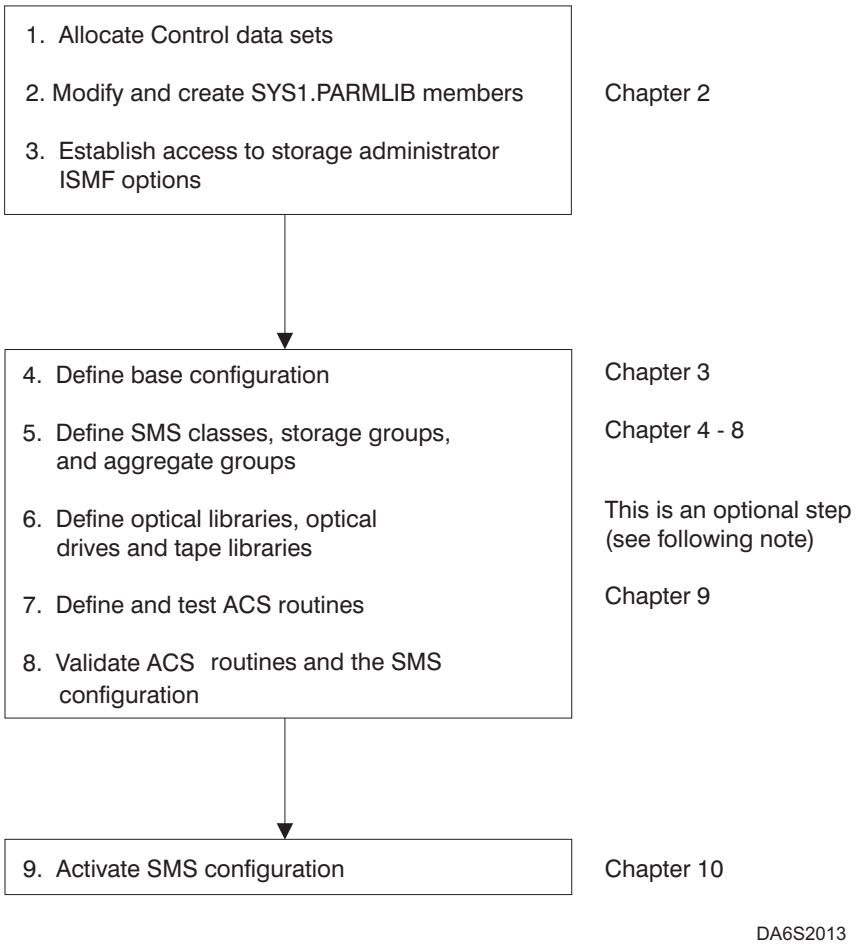


Figure 2. Activating an SMS Configuration

Related Reading: For information on step 6, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

When preparing for SMS, you need to perform the following steps:

1. Allocate control data sets, which contains information used by SMS.
2. Modify and create SYS1.PARMLIB members to identify SMS to all the systems in the SMS complex. These new members take effect when you IPL.

3. Establish access to the ISMF Primary Option Menu for Storage Administrators, which is shown in Figure 1 on page 2. This is the final preparation step.
4. Define the base configuration, which identifies the systems within the SMS complex.
5. Define the SMS classes and storage groups that you want SMS to assign to your data sets and objects, and the data sets that you want to assign to your aggregate groups.
6. Define optical libraries and drives and tape libraries, if you have them.
7. Define ACS routines to assign the SMS classes and storage groups. Then you test the routines.
8. Validate the ACS routines individually to check for errors. You should then validate the entire SMS configuration to check for errors that exist among its related parts.
9. Activate the valid SMS configuration.

See “Chapter 2. Preparing for the Storage Management Subsystem” on page 9 for a detailed description of each of these preparatory tasks.

Running SMS in a Parallel Sysplex Environment

An SMS complex consists of systems or *system groups* that share a common configuration. A Parallel Sysplex is made up of systems that share a cross-system coupling facility (XCF); you can run multiple SMS complexes within a Parallel Sysplex.

Restrictions:

- An SMS complex should not span sysplexes. All of the volumes in the SMS complex should be in the same Parallel Sysplex. The cross-system sharing functions, such as VSAM record-level sharing (RLS), partitioned data set extended (PDSE) sharing, RACF™ security and global shared resources (GRS) serialization, only work within the scope of a single Parallel Sysplex. These functions are not supported when the SMS complex extends beyond the Parallel Sysplex in which they are carried out.
- Do *not* set up multiple SMS complexes sharing the same DASD. This requires extra work to maintain the duplicate SMS configurations and can also create problems such as running out of disk space because one configuration cannot know about changes made to the other configuration, such as data set allocations and deletions, and storage group and volume status changes.

Basic Terms and Definitions

The following are some basic terms and definitions:

SMS complex
A system or a collection of systems that share a common configuration including a common active control data set (ACDS) and a common communication data set (COMMDS) pair. The SMS configuration now supports up to 32 system names, system group names, or both.

Parallel Sysplex
A collection of MVS™ systems in a multi-system environment supported by the cross-system coupling facility (XCF).

System name
The name of the system where an SMS operation is being performed.

System group
The system names within a Parallel Sysplex, excluding those systems in that sysplex, if any, that are individually defined in the SCDS. This support can be used even if the XCF is not active.

The Concept of System Grouping

In the MVS environment, a Parallel Sysplex is a collection of systems linked by closely coupled hardware facilities to process customer workloads. SMS system group name support allows you to specify a system group as a member of an SMS complex.

In SMS, the term *system group* is used instead of *Parallel Sysplex*. A system group consists of system names within a Parallel Sysplex, excluding those systems that are individually specified in the SMS base configuration. A system group name can represent multiple systems. This allows SMS to support more than eight systems per SMS complex while retaining the existing configuration format of the configuration data set.

Object storage groups, object backup storage groups, and optical libraries do not support system groups. If you have object storage groups, object backup storage groups or optical libraries in your configuration, the systems where they are to be enabled must be specified as individual system names.

A system group name matches a Parallel Sysplex name and refers to all systems defined as part of the Parallel Sysplex that are:

- Running the same SMS configuration
- Defined in the configuration using the name of the Parallel Sysplex to which they belong (that is, system group)
- Not defined in the configuration by their system names.

The system group name represents all the systems in the Parallel Sysplex which are not explicitly specified in the SCDS base configuration.

Figure 3 on page 7 provides a visual representation of the system grouping concept.

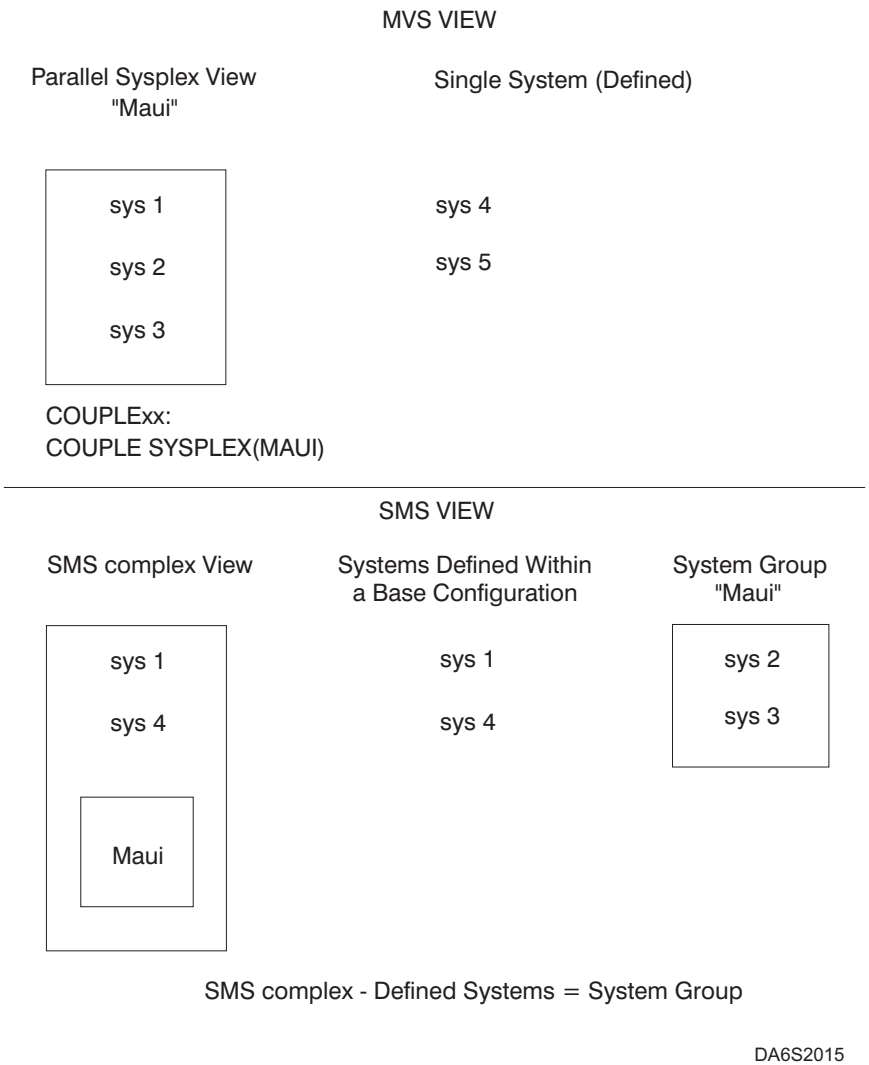


Figure 3. System Grouping Concept

In Figure 3, the SMS complex view shows that it is managing sys 1 and sys 4 as single systems, and Maui as a Parallel Sysplex. (sys 5 is not part of the SMS complex.)

The Parallel Sysplex (Maui) contains three system names: sys 1, sys 2, and sys 3. The single systems sys 1 and sys 4, as well as the Parallel Sysplex (Maui) are all defined in the SCDS. If you perform an SMS operation on Maui, it affects only sys 2 and sys 3 in the Parallel Sysplex. It does not affect sys 1 because it is defined separately in the SCDS.

In this example, the system group (sys 2 and sys 3) is represented by the Parallel Sysplex name, Maui.

Running SMS on Mixed DFSMS Releases

See *z/OS DFSMS Migration* for information on running SMS on mixed DFSMS releases.

Chapter 2. Preparing for the Storage Management Subsystem

Before you define and activate an SMS configuration, you need to perform the following preparatory steps:

- Allocate control data sets to contain your SMS configuration and to permit the systems in your complex to communicate with each other.
- Modify SYS1.PARMLIB, which contains three members that direct the initialization and activation of SMS.
- Establish access to the ISMF Primary Option Menu for Storage Administrators (shown in Figure 1 on page 2).

This chapter describes how to perform these preliminary steps so that you can begin defining a base configuration for an SMS configuration.

See *z/OS DFSMS: Implementing System-Managed Storage* for additional information regarding planning and implementing SMS.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for information on planning and preparing for SMS with object support, optical libraries, or tape libraries.

Allocating Control Data Sets

Before you can activate an SMS configuration, you need to allocate control data sets used by SMS and define their contents. Control data sets are virtual storage access method (VSAM) linear data sets that contain:

- Base configuration information
- SMS class, aggregate group, optical library, tape library, optical drive, and storage group definitions
- ACS routines

You can allocate control data sets with access method services or TSO/E commands. You can define and alter the contents of control data sets using ISMF. SMS uses three types of control data sets: a *source control data set* (SCDS), an *active control data set* (ACDS), and a *communications data set* (COMMDS).

Restriction: Do not name any of your SMS control data sets the single word 'ACTIVE'. SMS uses the single word 'ACTIVE' as a reserved word indicating the active configuration residing in the SMS address space. Naming an SMS control data set 'ACTIVE' results in errors.

Source Control Data Set (SCDS)

An SCDS contains an SMS configuration, which defines a storage management policy. You can define any number of SMS configurations each of which has its own SCDS. Then, you select one SMS configuration to be the installation storage management policy and make an active working copy of it in an ACDS.

Active Control Data Set (ACDS)

When you activate an SCDS, its contents are copied to an ACDS. The current ACDS contains a copy of the most recently activated configuration. All systems in an SMS complex use this configuration to manage storage. You can define any

© Copyright IBM Corp. 1986, 2002

number of SCDSs, but only one can be put in the ACDS. “Initializing SMS Through the IGDSMSxx Member” on page 14 explains how to specify the ACDS. You can define more than one IGDSMSxx member, each specifying a different ACDS, but you can use only one ACDS at a time.

Restriction: You cannot define or alter an ACDS. This also means that you cannot use an ACDS as an SCDS if the SCDS is lost.

You can modify the SCDS from which your current storage management policy was activated without disrupting operations, because SMS manages storage with a copy of the SMS configuration (an ACDS) rather than with the original (an SCDS). While SMS manages storage using an ACDS, you can:

- Create a backup copy of the SCDS
- Build a new SCDS
- Update the SCDS from which the ACDS was activated
- Modify any SCDS

Figure 4 shows the relationship among SCDSs and ACDSs in an installation.

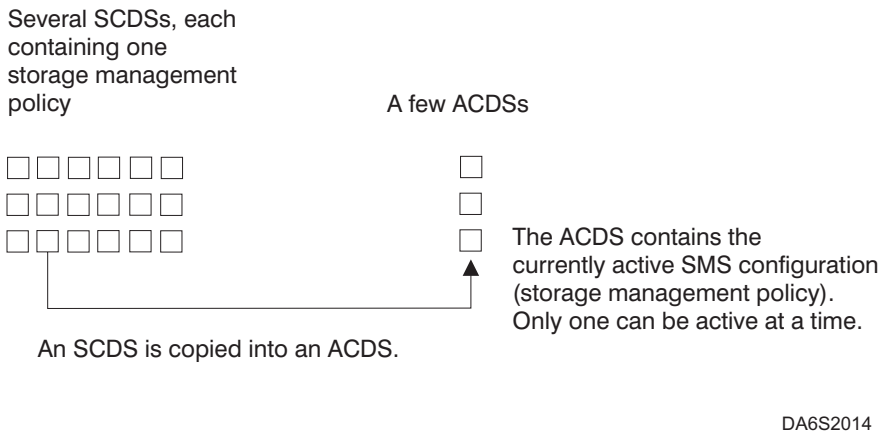


Figure 4. Relationship among SCDSs and ACDSs in an Installation

Communications Data Set (COMMDS)

The COMMDS serves as the primary means of SMS communication among systems in the SMS complex. The active systems in an SMS complex access the COMMDS for current SMS complex information.

The COMMDS contains the name of the ACDS containing the currently active storage management policy, the current utilization statistics for each system-managed volume, and other system information. You can define any number of COMMDSs, but only one can be active in an SMS complex.

Calculating the Size of Storage and Active Control Data Sets

Before you allocate control data sets, you need to estimate their size. When calculating the size of either an ACDS or an SCDS, you have to account for system and base configuration information, SMS class, aggregate group, storage group, optical library and drive, and tape library definitions.

The following formula can help you determine the size of a source or active control data set:

```
size (bytes) = 150000 +
               (14000 * SG) +
               (2312 * (MC + SC + DC + AG + CS)) +
               (12000 * (DRV + LIB + VOL))
```

where:

150000 bytes
represent fixed data fields. For example, if you have
- Five storage groups
- Four management classes
- Two storage classes
- Five data classes
- Two aggregate groups
- 6 optical libraries
- 24 optical drives
- 40 DASD volumes

the equation yields a value of 1088600 bytes for control data set allocation.

SG is the estimated number of storage groups

MC is the estimated number of management classes

SC is the estimated number of storage classes

DC is the estimated number of data classes

AG is the estimated number of aggregate groups

CS is the estimated number of cache sets in the base configuration (for record-level sharing (RLS) only)

DRV is the estimated number of optical drives in the SMS complex to be managed by SMS

LIB is the estimated number of optical and tape libraries in the SMS complex to be managed by SMS

VOL is the estimated number of DASD volumes in the SMS complex to be managed by SMS

If you are running SMS in a Parallel Sysplex environment with different releases of DFSMS, you must allocate your control data sets using calculations that are based on the highest DFSMS level. If you do not, your control data sets might be too small, because the storage requirements for classes and groups might be different between releases or new classes, groups, or other items might be added.

On a IBM 3380 or 9345 DASD each track can contain 40 KB (40960 bytes). On a IBM 3390 DASD each track can contain 48 KB (49156 bytes).

If your SCDS or ACDS is not large enough, you might receive SMS reason code 6068 when attempting to save its contents. When reason code 6068 occurs, allocate a new, larger control data set and copy your existing SCDS into your new SCDS, or your existing ACDS into your new ACDS. You can then delete the old SCDS or ACDS and use the new one.

Allocating a new SCDS or ACDS resolves the problem only when reason code 6068 is caused by a data set size problem. Because this reason code is returned when a system service called by data-in-virtual (DIV) fails, the error might have other causes. Messages returned to you or the system console can help determine the cause of the failure.

Restriction: DIV has a current size limit of 4 GB. Make sure you do not exceed this limit.

Calculating the Size of a COMMDS

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, and you should always review the COMMDS size when migrating from prior DFSMS releases. The following formula helps you determine the size of a COMMDS (VOL = estimated number of DASD volumes in the SMS complex to be managed by SMS):

COMMDS size (bytes) =8192 + (588 * VOL)

For example, if you have 40 DASD volumes in the SMS complex, you need to allocate 31712 bytes for the COMMDS.

Selecting Volumes for Control Data Sets

SMS control data sets can be either SMS-managed or non-SMS-managed. Initially you should ensure that your control data sets have a volume count of one. The volume count can be either explicitly specified, implied by the number of volume serials provided, or derived from the data class assigned to the data set (see “Chapter 7. Defining Data Classes” on page 111 for more information). If you give an SMS control data set a volume count that is greater than the number of volumes on which it actually resides you might receive messages IEF244I and IEF489I when you attempt to activate it.

If you have a multivolume SCDS which you are activating into a single volume ACDS, you might receive an error because the ACDS is not large enough and volumes cannot be dynamically added to it. To bypass this problem, you need to create a new multivolume ACDS and then activate the ACDS and SCDS simultaneously using the SETSMS command. See “Changing Storage Management Subsystem Parameters” on page 173 for further information on this command.

If your SMS complex includes more than 16 systems, be sure that the ACDS and COMMDS are accessible to every system in the complex. Define your control data sets on volumes which are capable of being attached to more than 16 systems, such as IBM RAMAC™ Virtual Array volumes.

Allocating an SCDS

The following access method services job allocates a 6-track SCDS.

```
//STEP  EXEC PGM=IDCAMS
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER(NAME(SMS.SCDS1.SCDS) LINEAR VOL(SMSV01) -
  TRK(6 6) SHAREOPTIONS(2,3)) -
  DATA(NAME(SMS.SCDS1.SCDS.DATA))
/*
```

This job creates a VSAM linear data set named SMS.SCDS1.SCDS. You can combine the DEFINE commands for all your allocations into the same job step, but this example shows only one for purposes of illustration. After allocating an SCDS, you define its contents through ISMF dialogs.

You should allocate an SCDS on a device shared by all systems in the SMS complex. If you allocate an SCDS on a device that is not shared by all the systems, then you can activate the SCDS only from systems that have access to it.

You should specify the REUSE option when you define an SCDS to avoid running into space problems (SMS reason code 6068) as result of subsequent SCDS updates, or IMPORT/EXPORT functions.

See *z/OS DFSMS Access Method Services* for information on using access method services commands.

Allocating an ACDS

The following access method services job allocates a 6-track ACDS.

```
//STEP  EXEC PGM=IDCAMS
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER(NAME(SMS.ACDS1.ACDS) LINEAR VOL(SMSV02) -
  TRK(6 6) SHAREOPTIONS(3,3)) -
  DATA(NAME(SMS.ACDS1.ACDS.DATA))
/*
```

This job creates a VSAM linear data set named SMS.ACDS1.ACDS.

An ACDS must reside on a shared volume, accessible from all systems in the SMS complex. To ease recovery in case of failure, the ACDS should reside on a different volume than the COMMDS. Also, you should allocate a spare ACDS on a different shared volume. “Chapter 12. Recovering Storage Management Subsystem Information” on page 215 provides additional information on the backup and recovery of control data sets.

You create the contents of an ACDS by activating a valid SCDS. The distinction between a valid SCDS and one that is not valid is described in “Defining the Base Configuration” on page 27. The control data set (ACDS or COMMDS) must reside on a volume that is not reserved by other systems for a long period of time because the control data set (ACDS or COMMDS) must be available to access for SMS processing to continue.

You should specify the REUSE option when you define an ACDS to avoid running into space problems (SMS reason code 6068) as result of subsequent ACDS updates, or IMPORT/EXPORT functions.

Allocating a COMMDS

The following access method services job allocates a 1-track COMMDS;

```
//STEP  EXEC PGM=IDCAMS
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE CLUSTER(NAME(SMS.COMMDS1.COMMDS) LINEAR VOL(SMSVOL) -
  TRK(1 1) SHAREOPTIONS(3,3)) -
  DATA(NAME(SMS.COMMDS1.COMMDS.DATA))
/*
```

This job creates a VSAM linear data set named SMS.COMMDS1.COMMDS.

The COMMDS must reside on a shared volume accessible from all systems in the SMS complex. To ease recovery in case of failure, the COMMDS should reside on a different volume than the ACDS. Also, you should allocate a spare COMMDS on a different shared volume. “Chapter 12. Recovering Storage Management Subsystem Information” on page 215 provides additional information on the backup and recovery of control data sets. The control data set (ACDS or COMMDS) must reside on a volume that is not reserved by other systems for a long period of time because the control data set (ACDS or COMMDS) must be available to access for SMS processing to continue.

We recommend that you use SHAREOPTIONS(3,3) when allocating an ACDS. This allows full authority to read from and write to an ACDS from any system. The ACDS and COMMDS must be accessed from all systems in the complex simultaneously.

Modifying the SYS1.PARMLIB Data Set

The IGDSMS*xx*, IEASYS*yy*, and IEFSSNxx members of SYS1.PARMLIB direct the initialization and activation of SMS. IGDSMS*xx* provides initialization parameters to SMS. The SMS=*xx* parameter of IEASYS*yy* indicates the name of the SYS1.PARMLIB member IGDSMS*xx* that is used for initialization. For example, if SMS=01 in IEASYS*yy*, then the IGDSMS01 member of SYS1.PARMLIB is used during initialization. The SMS entry in IEFSSNxx identifies SMS to z/OS.

This section explains how to create an IGDSMS*xx* member in SYS1.PARMLIB and describes two methods to define SMS to z/OS through IEFSSNxx. For additional information, see *z/OS MVS Initialization and Tuning Guide*.

Initializing SMS Through the IGDSMSxx Member

For each system in the SMS complex, you must create an IGDSMS*xx* member in SYS1.PARMLIB. The IGDSMS*xx* member contains SMS initialization control information, where *xx* is any value allowed by the naming conventions for SYS1.PARMLIB members. It has a default value of 00.

Every SMS system must have an IGDSMS*xx* member in SYS1.PARMLIB that specifies a required ACDS and COMMDS pair. This ACDS and COMMDS pair is used if the COMMDS of the pair does not point to another COMMDS.

If the COMMDS points to another COMMDS, the referenced COMMDS is used. This referenced COMMDS might contain the name of an ACDS that is different from the one specified in the SYS1.PARMLIB member. If so, the name of the ACDS is obtained from the COMMDS rather than from the IGDSMS*xx* member in SYS1.PARMLIB to ensure that the system is always running under the most recent ACDS and COMMDS.

If the COMMDS of the ACDS and COMMDS pair refers to another COMMDS during IPL, it means a more recent COMMDS has been used. SMS uses the most recent COMMDS to ensure that you cannot IPL with a down-level configuration.

The data sets that you specify for the ACDS and COMMDS pair must be the same for every system in an SMS complex. Whenever you change the ACDS or COMMDS, update the IGDSMS*xx* member of SYS1.PARMLIB for every system in the SMS complex so that it specifies the same data sets.

The IGDSMS*xx* member contains all the parameters shown in Table 1. These parameters direct the SMS initialization process and specify the names of the ACDS and COMMDS.

Table 1. Parameters of the IGDSMSxx Member of SYS1.PARMLIB

SMS	**ACDS(***dsname***)** **COMMDS(***dsname***)** **[ACSDEFAULTS({YES|NO})]** **[ASID({asid|*})]** **[BMFTIME({nnn|3600})]** **[CACHETIME({nnn|3600})]** **[CF_TIME({nnn|3600})]** **[COMPRESS({TAILORED|GENERIC})]** **[DB2SSID(***ssid***)]** **[DEADLOCK_DETECTION({iiii|15,kkkk|4})]** **[DESELECT({event[,event][,...]|ALL})]** **[DINTERVAL({nnn|150})]** **[DSNTYPE({LIBRARY|PDS})]** **[HSP_SIZE(***nnn***)]** **[INTERVAL({nnn|15})]** **[JOBNAME({jobname|*})]** **[LRUCYCLES({nnn|240})]** **[LRUTIME({nnn|15})]** **[MAXLOCKS({max|0},{incr|0})]** **[OAMPROC(***procname***)]** **[OAMTASK(***taskid***)]** **[OVRD_EXPDT({YES|NO})]** **[PDSESHARING({NORMAL|EXTENDED})]** **[PDSE_MONITOR({YES|NO}[,***interval***][,***duration***])]** **[REVERIFY({YES|NO})]** **[RLSINIT({NO|YES})]** **[RLS_MAXCfFeatureLevel({A|Z})]** **[RLS_MAX_POOL_SIZE({nnnn|100})]** **[RLSTMOUT({nnn|0})]** **[SELECT({event[,event][...]|ALL})]** **[SIZE(***nnn***{K|M})]** **[SMF_TIME({YES|NO})]** **[SYSTEMS({32|8})]** **[TRACE({OFF|ON})]** **[TRACEEXIT(***user_trace_exit***)]** **[TYPE({ALL|ERROR})]** **[USE_RESOWNER({YES|NO})]**

ACDS, COMMDS, REVERIFY, and ACSDEFAULTS need to be the same for all your systems. The rest of the parameters can be different.

Related Reading: For more detailed descriptions of the parameters listed in Table 1 on page 15, see *z/OS MVS Initialization and Tuning Guide* and *z/OS MVS System Commands*.

Required Keywords

Blanks are permitted between SMS keywords.

SMS
Identifies the member as a repository of SMS initialization control information.

ACDS(*dsname*)
identifies the name of the data set containing the active configuration. If you omit *dsname*, the operator is prompted for a value.

COMMDS(*dsname*)
identifies the name of the COMMDS. If you omit *dsname*, the operator is prompted for a value.

Optional Keywords

ACSDEFAULTS({YES|NO})
indicates whether SMS initializes the following ACS routine variables from an additional call to the Resource Access Control Facility (RACF), a component of the Security Server for z/OS:

&APPLIC
&DEF_DATACLAS
&DEF_MGMTCLAS
&DEF_STORCLAS

If you specify **NO**, these variables have no values associated with them. The default value for ACSDEFAULTS is **NO**.

Exceptions: The ACSDEFAULTS keyword is not applicable when USE_RESOWNER(NO) keyword is also specified. The ACSDEFAULTS keyword is not applicable for OAM.

ASID({*asid*|*})
limits tracing to a certain address space or permits it for all address spaces. Specify * if you want all address spaces traced (providing SMS tracing is activate). This is the default. You can enter up to 4 digits for the ASID keyword. If you leave off the leading zeroes, they are inserted.

BMFTIME({*nnn*|3600})
specifies the number of seconds between recording SMF record type 42, subtype 1 records for PDSE (where the buffer hits are I/O requests which did not require actual disk I/O). You can specify a value from 1-86399 (23 hours, 59 minutes, 59 seconds). The default is 3600 (1 hour).

CACHETIME({*nnn*|3600})
specifies the number of seconds between recording SMF records for device cache use. The **CACHETIME** parameter applies only to the volumes behind an IBM 3990 Storage Control with cache unit. You can specify a value from 1-86399 (23 hours, 59 minutes, 59 seconds). The default is 3600 (1 hour).

CF_TIME({*nnn*|3600})
indicates the number of seconds between recording SMF records for the coupling facility (both cache and lock). You can specify a value from

1-86399 (23 hours, 59 minutes, 59 seconds). The default is 3600 (1 hour). This keyword sets the interval time for the following SMF 42 subtypes:

SUBTYPE 15 Coupling facility storage class average response time
SUBTYPE 16 Coupling facility data set average response time
SUBTYPE 17 Coupling facility lock structure activity
SUBTYPE 18 Coupling facility cache partition summary
SUBTYPE 19 SMSVSAM least recently used statistics summary

COMPRESS({TAILORED|GENERIC})
Specifies the type of compression to be used for the data set.

TAILORED specifies that the data set is eligible for compression specifically tailored to the data set. A tailored dictionary is built, using the initial data written to the data set, and imbedded into the data set. The dictionary is used to compress or expand data written to or read from the data set. This type of compression applies only to sequential data sets, not to VSAM KSDSs.

To convert an existing DBB-based compressed data set to use tailored compression, first set the COMPRESS parameter to TAILORED in the IGDSMSxx parmlib member. Use IEBGENER, ICEGENER, REPRO, or any QSAM or BSAM application to copy the DBB-based data set to a new data set that meets compression requirements.

GENERIC specifies that the data set be compressed using generic Dictionary Building Block (DBB) compression. The dictionary is derived from a defined set of compression algorithms in data set SYS1.DBBLIB. This is the default.

Requirement: Use tailored compression only when all systems in the SMS complex have been converted to DFSMS/MVS V1R4 or higher, and when there is no need to revert to a prior release level for local recovery or remote recovery with Aggregate Backup and Recovery Support (ABARS).

DB2SSID(*ssid*)
identifies either the name of the DB2® subsystem used by OAM for object storage, or a group attachment name if data sharing is being used. If you specify a group attachment name, the DB2 startup ECB is ignored by DB2, so DB2 must be initialized before OAM is started.

If OAM object support and DB2 are not needed for OAM's object directories, object data tables, and configuration database, this parameter is not necessary and should not be specified. The *ssid* can be from one to four characters and there is no default.

DEADLOCK_DETECTION({iiii|15,kkkk|4})
specifies the deadlock detection intervals used by the DFSMSdfp Storage Management Locking Services. The first subparameter is the local deadlock detection cycle and specifies the interval in seconds for detecting deadlocks within a system. The second subparameter is the global deadlock detection cycle and specifies the interval for detecting deadlocks between systems. The value is specified as the number of local detection cycles that occur before global deadlock detection is initiated.

iiii one to four digit numeric value in the range 1-9999 that specifies the length in seconds of the local deadlock detection interval. The default is 15 seconds.

kkkk one to four digit numeric value in the range 1-9999 that specifies the number of local deadlock cycles that must complete before global deadlock detection occurs. The default is 4.

DESELECT({event[,event][,...]|ALL})

deletes items from the list of events to be traced (if SMS tracing is active). DESELECT has no default. If you specify events that conflict in SELECT and DESELECT, the keyword that appears last has final authority.

The events that you can specify on SELECT and DESELECT are:

MODULE	SMS module entry or exit
DSTACK	SMS Data Set Stacking Service
SMSSJF	SMS/SJF interfaces
SMSSSI	SMS/SSI interfaces
ACSINT	ACS services interfaces
OPCMD	Operator commands
CONFC	Configuration change
CDSC	Control data set changes
CONFS	SMS Configuration services
MSG	SMS Message services
ERR	SMS Error recovery and recording services
CONFR	Return data from an active configuration
CONFA	Activate a new configuration
ACSPRO	Perform ACS processing
IDAX	SMS interpreter/dynamic allocation
DISP	SMS disposition processing exit
CATG	SMS catalog services
VOLREF	SMS VOLREF services
SCHEDP	SMS scheduling services (preallocate catalog orientation)
SCHEDS	SMS scheduling services (system select)
VTOCL	SMS VTOC/data set services (allocate existing data set)
VTOCD	SMS VTOC/data set services (delete existing data set)
VTOCR	SMS VTOC/data set services (rename existing data set)
VTOCC	SMS VTOC/data set services (allocate new data set)
VTOCA	SMS VTOC/data set services (add a volume to a data set)
RCD	SMS Recording services
DCF	SMS device control facility
DPN	SMS device pool name select subsystem interface
TVR	SMS tape volume record update facility
ALL	All of the above options

DINTERVAL({*nnn*|150})
specifies the number of seconds SMS allows to elapse before it reads device statistics, which SMS uses to manage hardware usage and maximize efficiency. The **DINTERVAL** parameter applies only to the volumes behind an IBM 3990 Storage Control with cache unit. You can specify a value from 1-999 (16 minutes, 39 seconds). The default is 150.

DSNTYPE({LIBRARY|PDS})
specifies the installation default for data sets allocated with directory space but without a data set type specified. If the data set type is **PDS**, the default is a partitioned data set format; if the data set type is **LIBRARY**, the default is a PDSE (partitioned data set extended) format.

HSP_SIZE(*nnn*)
specifies the size, in megabytes, of the hiperspace that is used to cache PDSE member pages. You can specify a hiperspace size up to 512 MB. The valid range for HSP_SIZE is 0 to 512. If you specify 0, the hiperspace is not created. To create a hiperspace, your system must have expanded storage available. If expanded storage is available and you do not specify a value for HSP_SIZE, the default is 256 MB or half of expanded storage, whichever is less.

INTERVAL({*nnn*|15})
specifies the *synchronization time interval* of the system, which is the number of seconds between system checks of the COMMDS for news about SMS configuration changes from other systems in the SMS complex. You can specify a value from 1-999 (16 minutes, 39 seconds). The default is 15.

Recommendation: OAM depends on the time interval specified with this INTERVAL keyword in order to be notified of an SCDS activation. Therefore, be sure to specify an appropriate value for this keyword when it is used by other address spaces that are also dependent on the SMS configuration.

JOBNAME({*jobname*|*})
limits tracing to a certain job or permits tracing on all jobs. The default is *****, to trace all jobs. This keyword supports objects or tape libraries.

LRUCYCLES(nnn|240)
Specifies the maximum number of times (5 to 240 cycles) that the buffer management facility (BMF) least recently used (LRU) routine will pass over inactive buffers before making them available for reuse. While this parameter sets the **maximum** value, BMF will dynamically change the **actual** number of times it passes over inactive buffers.

LRUCYCLES is related to LRUTIME. A change to the LRUCYCLES value introduced by this parameter will take effect on the next execution of the LRU routine.

The default is 240.

Recommendation: Most installations use the default value. In some very high data rate situations you might tune this value. Monitor the SMF type 42 subtype 1 record to determine the amount of caching activity in the BMF data space.

Related Reading: For more information about the buffer management statistics recorded in SMF record type 42, see *z/OS MVS System Management Facilities (SMF)*.

LRUTIME(nnn|15)

Specifies the number of seconds (5 to 60) that the buffer management facility (BMF) will wait between calls to the BMF data space cache LRU (least recently used) routine. The LRU routine releases inactive buffers in the BMF data space that are used to cache PDSE (partitioned data set extended) directory data.

LRUTIME is related to LRUCYCLES. A change to the LRUTIME value introduced by this parameter will take effect on the next execution of the LRU routine.

The default is 15.

Recommendation: Most installations use the default value. In some very high data rate situations you might tune this value. Monitor the SMF type 42 subtype 1 record to determine the amount of caching activity in the BMF data space.

Related Reading: For more information about the buffer management statistics recorded in SMF record type 42, see *z/OS MVS System Management Facilities (SMF)*.

OAMPROC(*procname*)

identifies the procedure name that starts the OAM address space when SMS is initialized. You must specify this keyword if you want the OAM address space started during IPL. The procedure name can be from one to eight characters and there is no default.

OAMTASK(*taskname*)

specifies the task ID that is used to start the OAM address space. **OAMTASK** is optional; if it is specified without **OAMPROC** it is ignored. If omitted, the task ID defaults to the procedure name specified in **OAMPROC**. OAM keywords take effect only if you initialize SMS at IPL; otherwise they are not saved. The task ID can be from one to eight characters.

OVRD_EXPDT({YES|NO})

allows for the override of an expiration date when an unexpired SMS-managed DASD data set is deleted. If you specify NO, expiration dates are honored unless specific action is taken to override them. If you specify YES, any expiration date specified for the data set is overridden when deletion of an SMS-managed DASD data set is attempted through JCL, SVC 99, IEHPROGM or the SCRATCH macro. YES should only be used when management class cannot be used to override expiration dates. It is intended to help with data sets which are converted from tape to DASD since expiration date is more commonly used with tape.

PDSESHARING({NORMAL|EXTENDED})

specifies whether PDSEs can be shared by all systems within a Parallel Sysplex for input only or for input and output. A value of **NORMAL** means that PDSEs can be shared by all the systems in a Parallel Sysplex but *only* for input. A value of **EXTENDED** means that PDSEs can be shared by all the systems in a Parallel Sysplex for input *and* output. The default is **NORMAL**.

PDSE_MONITOR({YES|NO}[,*interval*[,*duration*]])

specifies how the processing for the PDSE monitor is started or modified.

To turn monitor processing on, specify **YES**. To turn monitor processing off, specify **NO**. The default is **YES**.

For *interval*, specify the number of seconds between successive scans of the monitor. If you do not specify a value, the interval is unchanged except at IPL, when it is set to 60. For *duration*, specify the number of seconds a possible error condition must exist before it is treated as an error. If you do not specify a value, the duration is unchanged except at IPL, when it is set to 15.

REVERIFY({YES|NO})
specifies whether SMS verifies authorization only at job interpretation time or at both job interpretation and job execution time. A value of **NO** instructs SMS to verify a user's authority to allocate a new data set, to use a storage class, and to use a management class at job interpretation time only. A value of **YES** instructs SMS to perform the verification at both job interpretation and job execution time. JES2 environments ignore the **REVERIFY** keyword. It has a default value of **NO**.

[RLSINIT({NO|YES})]
specify YES if you want the SMSVSAM address space started as part of system initialization or the V SMS,SMSVSAM,ACTIVE command. This value applies only to the system accessed by the parmlib member and is acted upon when SMSVSAM is next started. The default is **NO**.

[RLS_MAXCfFeatureLevel({A|Z})]
specifies the method that VSAM RLS uses to determine the size of the data that is placed in the CF cache structure.

If you specify **A**, caching proceeds using the RLS CF Cache Value keyword characteristics that are specified in the SMS data class that is defined for the VSAM sphere.

If you do not specify a value, or if you specify **Z**, then only VSAM RLS data that have a Control Interval (CI) value of 4K or less are placed in the CF cache structure. The default is **Z**.

Restrictions:

- If **A** is specified for the RLS_MaxCfFeatureLevel parameter, systems lower than V1R3 will not be able to connect to the CF cache structure.
- If a lower-level system is the first system activated in the sysplex, RLS_MaxCfFeatureLevel defaults to **Z**, and all systems will be able to connect to the CF cache structure.
- If the SETSMS command is used to change the RLS_MaxCfFeatureLevel value to **A** on a mixed-level system, the command is rejected and message IGW500I is issued.

RLS_MAX_POOL_SIZE({nnnn|100})
specifies the maximum size in megabytes of the SMSVSAM local buffer pool. SMSVSAM attempts to not exceed the buffer pool size you specify, although more storage might be temporarily used. Because SMSVSAM manages buffer pool space dynamically, this value does not set a static size for the buffer pool.

Use SMF type 42, subtype 19 records to help you determine the maximum size of the SMSVSAM local buffer pool.

You can specify a two to four-digit numeric value, with 10 as the minimum value. If you specify a value less than 10, the field is set to 10. If you specify a value greater than 1500, SMSVSAM assumes there is no maximum limit. We recommend that you limit the size of the local buffer pool. The default is 100.

RLSTMOUT({nnn|0})
specifies the maximum time, in seconds, that a VSAM RLS request must wait for a required lock before the request is assumed to be in deadlock. You can specify a value between 0 to 9999 (in seconds). A value of 0 means that the VSAM RLS request has no time out value and the request will wait for as long as necessary to obtain the required lock.

RLSTMOUT can be specified only once in a sysplex and applies across all systems in the sysplex.

The default is 300.

SELECT({*event*[, *event*][,...]|ALL})
adds items to the list of events to be traced (if SMS tracing is active). The default is **ALL**. See **DESELECT** for a list of valid events.

SIZE(*nnn*{K|M})
specifies the size of the SMS trace table in bytes. When the unit is **K**, the value can range from 0K to 255000K, and it is rounded up to the nearest 4K unit. When the unit is **M**, the value can range from 0M - 255M. If you specify a value of 0, no tracing is performed. The default is 128K.

SMF_TIME({YES|NO})
YES indicates that the following SMF type 42 records are created at the SMF interval time, and that all of the indicated records are synchronized with SMF and RMF™ data intervals:

SUBTYPE 1	Buffer management statistics
SUBTYPE 2	Cache control unit statistics (IBM 3990-3, 3990-6, IBM RVA, and IBM ESS)
SUBTYPE 15	Coupling facility storage class average response time
SUBTYPE 16	Coupling facility data set average response time
SUBTYPE 17	Coupling facility lock structure activity
SUBTYPE 18	Coupling facility cache partition summary
SUBTYPE 19	SMSVSAM least recently used statistics summary

DFSMS creates the specified SMF record at the end of the interval period and SMF sends the event notification signal. If YES is specified, this subparameter overrides the following subparameters: BMFTIME, CACHETIME, CF_TIME. The default is YES.

Related Reading: For more information on SMF, see *z/OS MVS System Management Facilities (SMF)*.

SYSTEMS({32|8})
indicates whether the system is running in compatibility mode (eight name limit) or 32-name mode. The default is **8**.

SYSTEMS(8) specifies that a maximum of eight system names, system group names, or both, can be specified in the SMS configuration. This indicates that the system is running in compatibility mode and can share configurations (SCDSs or ACDSs) and COMMDSs with systems that are running down-level releases of DFSMS or DFP. Essentially, the system continues to operate as it has in the past.

SYSTEMS(32) specifies that a maximum of 32 system names, system group names or both can be specified in the SMS configuration. This indicates the system is not running in compatibility mode.

Tip: The ACDS, SCDS and COMMDS might not be shared with the systems that are running down-level releases of DFSMS or DFP or that are running in compatibility mode.

TRACE({OFF|ON})
indicates whether the SMS trace facility is activated. The default is **ON**. The SMS trace facility records trace records in the SMS address space.

See Also: For more information about the TRACE parameter, see *z/OS DFSMSdfp Diagnosis Reference*.

TRACEEXIT(*user_trace_exit***)**
specifies the name of your trace exit routine. You can specify an alphanumeric value of 1 - 8 characters, starting with an alphabetic character. Your value must represent a valid module name. The default value is blank. This interface is usually used only for diagnostic purposes under the guidance of the service organization and therefore is not documented here.

TYPE({ALL|ERROR})
specifies whether SMS traces all events or only errors. The default is **ERROR**.

USE_RESOWNER({YES|NO})
indicates whether construct authorization checking is done using the RESOWNER value, which is based on the high-level qualifier of the data set name, or using the data set allocator user ID. If you specify **NO**, the RESOWNER value is not extracted and the allocator user ID is used. The default value for USE_RESOWNER is **YES**.

Defining SMS to z/OS

Before you can activate SMS, z/OS must recognize SMS as a valid subsystem. You can define SMS to z/OS with IEFSSNxx.

Defining SMS to z/OS through IEFSSNxx

You can define SMS to z/OS through IEFSSNxx in one of two ways. The first method involves placing a record for SMS in the existing IEFSSNxx member that is used for the next initial program load (IPL). The format for the SMS entry (record) is shown in “Writing the SMS Record” on page 24.

The second method involves creating a new IEFSSNxx member that contains only the record for SMS. To identify the IEFSSNxx member containing the SMS entry, you must update the IEASYS*yy* member that you use to IPL. In the SSN parameter of IEASYS*yy*, the suffix identifying the IEFSSNxx member that contains the SMS entry should appear first, because SMS must be active before starting any other subsystem. For example, in:

```
SSN=(00,01,...)
```

the `00` identifies IEFSSN00 as the first member, and `01` identifies IEFSSN01 as the second member, and so forth. You can specify additional members to define more subsystems. For example:

```
SSN=(00,01,02,03).
```

You have the option of manually starting SMS with a command or automatically starting it at future IPLs. When you are first preparing for SMS, you might want to omit the IGDSSIIN module name from the SMS entry. This defines SMS to z/OS at the next IPL but does not automatically start SMS. After you IPL the system, SMS is defined as a subsystem and you can then start it manually.

Writing the SMS Record

To define SMS to z/OS, you must place a record for SMS in an IEFSSNxx member. You can code an IEFSSNxx member with keyword or positional parameters, but not both.

Recommendation: Use keyword parameters to code an IEFSSNxx member. You must use the same format for all systems on which you are running SMS.

The keyword syntax you can use to define SMS in IEFSSNxx is:

```
SUBSYS SUBNAME(SMS)
[INITRN(IGDSSIIN) [INITPARM('ID=yy,PROMPT=NO)]]
                                          YES)]]
                                          DISPLAY)]]
```

If you choose to use positional instead of keyword parameters, the positional format of the SMS definition in IEFSSNxx is:

```
SMS[,[IGDSSIIN][,'ID=yy[,PROMPT=NO]]
                                    YES]]
                                    DISPLAY]]
```

Required Keywords:

SMS identifies and defines the Storage Management Subsystem to z/OS.

Optional Keywords:

IGDSSIIN identifies the subsystem initialization routine IGDSSIIN for SMS. If you include it in the SMS entry, SMS is automatically started during every future IPL. If you omit it from the SMS entry, SMS is defined as a valid subsystem to z/OS, but does not automatically start during future IPLs.

ID={*xx*|00} specifies the SYS1.PARMLIB member used for initialization in two special cases:

- When the value specified in the SMS parameter of IEASYS*yy* does not correspond to any IGDSMS*xx* and the default IGDSMS00 does not exist.
- When initialization of software controlling PDSEs fails.

To use PDSEs, SMS and the PDSE address space must be active. SMS allows you to allocate PDSEs and the PDSE address space allows you to open, read, write, and update PDSEs. If the PDSE address space initializes, the IGDSMS*xx* member used for SMS initialization is determined by the SMS=xx parameter in IEASYS*yy*. If the PDSE address space fails, the IGDSMS*xx* member used for SMS initialization is determined by the ID=xx parameter in IEFSSNxx. The SMS parameters should be the same unless you want different parameters for SMS when PDSEs are not available.

You should specify the same value in SMS=*xx* in IEASYS*yy* and in ID=*xx* in IEFSSNxx to avoid confusion.

If you specify both **ID** and **PROMPT**, enclose them in single quotation marks and separate them with a comma.

PROMPT={DISPLAY|YES|NO}
indicates how much control you want the operator to have during the rest of SMS initialization. DISPLAY displays the contents of the IGDSMS*xx* member, but the operator cannot modify them. YES displays the contents of the IGDSMS*xx* member and prompts the operator to modify the parameters in the IGDSMS*xx* member. The modifications take effect for the current IPL, but the contents of the IGDSMS*xx* SYS1.PARMLIB member do not change. NO, which is the default, neither displays the parameters nor allows the operator to modify them.

If you specify more than one keyword, enclose them in single quotation marks and separate them with a comma.

Starting the SMS Address Space

When you have completed preparations and are ready to start SMS, use the T SMS=*xx* command, where *xx* identifies IGDSMS*xx* as the SMS initialization member. The T SMS=*xx* command is an abbreviation for the SET SMS=*xx* command, discussed in “Step Two: Prepare One System” on page 171. To eliminate confusion with the SETSMS operator command, the abbreviated T SMS=*xx* form of the SET SMS=*xx* command is used throughout the remainder of this manual.

When you have sufficiently tested your operations and are ready to have SMS automatically started at future IPLs, add the IGDSSIIN module name to the SMS entry.

Here are some examples of the SMS record:

- The following SMS record defines SMS to z/OS without starting SMS at IPLs:
  ```
  SUBSYS SUBNAME(SMS)
  ```
- The following SMS record defines SMS to z/OS and starts SMS at future IPLs:
  ```
  SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)
  ```

  The system uses the default values of **ID** and **PROMPT**. IGDSMS00 specifies initialization information, and the operator has no control over the rest of SMS initialization.
- The following SMS record allows the operator to modify SMS initialization:
  ```
  SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)
         INITPARM(',PROMPT=YES')
  ```

  The system uses the default value of **ID**, which identifies IGDSMS00 as containing initialization information. The **PROMPT** parameter requests that SMS display IGDSMS00, so that the operator can modify the parameters in IGDSMS00.
- The following SMS record initializes SMS using IGDSMS01:
  ```
  SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)
         INITPARM('ID=01,PROMPT=DISPLAY')
  ```

  The **PROMPT** parameter requests that the contents of IGDSMS01 be displayed, but the operator cannot modify them.

Accessing the Storage Administrator Primary Option Menu

The first time you select ISMF, you get the ISMF Primary Option Menu for *end users*. To get the ISMF Primary Option Menu for *storage administrators* (which is shown in Figure 1 on page 2), select option 0, `ISMF PROFILE`. Within the ISMF Profile Option Menu, select option 0, `USER MODE`, and press ENTER. You get the User Mode Entry panel, where you indicate that you want the storage administrator Primary Option Menu for all future ISMF sessions. To do this, select option 2 on the User Mode Entry panel. After changing the user mode, you must exit ISMF and then return to it to view the Primary Option Menu for Storage Administrators.

"Chapter 13. Protecting the Storage Management Subsystem" on page 219 explains how to prevent end users from gaining access to the storage administrator Primary Option Menu through the ISMF PROFILE option. The reason for restricting access to the Primary Option Menu for Storage Administrators is to prevent unauthorized users from performing storage administrator tasks.

Planning for VSAM Record-Level Sharing

Planning for and installing VSAM record-level sharing requires coordination with system hardware and software groups. The following planning tasks are described fully in "Chapter 14. Administering VSAM Record-Level Sharing" on page 235:

- "Determining Hardware Requirements" on page 235
- "Determining Applications That Can Use VSAM RLS" on page 236
- "Ensuring Same Systems Connectivity" on page 237
- "Planning for Availability" on page 238
- "Defining Sharing Control Data Sets" on page 239
- "Defining CF Cache Structures" on page 241
- "Defining the CF Lock Structure" on page 242
- "Modifying the SYS1.PARMLIB IGDSMSxx Member" on page 247
- "Establishing Authorization for VSAM RLS" on page 247

Chapter 3. Creating the Base Configuration

The first thing that you define in an SCDS is the base configuration. A base configuration contains installation defaults, such as a default management class, and identifies the systems to which the SMS configuration applies.

This chapter describes the base configuration and explains how to define its contents using the ISMF control data set application.

Planning the Base Configuration

You need to determine the system names and system group names that you want to specify in the SCDS. These systems and system groups constitute the SMS complex.

Before defining the base configuration, consider what you want to do with system-managed data sets that do not have a management class. You can specify a default management class for these data sets in the base configuration. You can create the management class any time before validating the SCDS. Defining the management classes is described in “Chapter 5. Defining Management Classes” on page 69.

If the management class ACS routine does not determine a management class for a data set, DFSMShsm™ processes the data set using the default management class, if one exists. If you do not specify a default management class in your base configuration, DFSMShsm uses its own defaults.

If the management class ACS routine does not determine a management class for an *object*, the OSREQ request fails. This request can be OSREQ STORE, CHANGE, or OSMC class transition processing.

See *z/OS DFSMS: Implementing System-Managed Storage* for additional planning information.

Defining the Base Configuration

After allocating an SCDS, you can define a base configuration. By selecting option 8 from the ISMF Primary Option Menu for Storage Administrators, you can invoke the Control Data Set (CDS) Application Selection panel shown in Figure 5 on page 28.

© Copyright IBM Corp. 1986, 2002

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                            CDS APPLICATION SELECTION
Command ===>

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

Select one of the following Options:

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

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

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

Figure 5. CDS Application Selection

On this panel, you can specify the name of the SCDS that is to contain the base configuration. ISMF primes the CDS Name field with the last used SCDS name.

After specifying an SCDS, select option 1, Display, and press Enter to see the base configuration. You can use this option to look at the base configuration of the current active SMS configuration by specifying 'ACTIVE' in the CDS Name field.

If you want to define a base configuration, select option 2, Define, and press Enter, to see the SCDS Base Define panel shown in Figure 6 on page 29.

If you want to alter an SCDS base configuration that you defined previously, select option 3, Alter, from the CDS Application Selection panel and press Enter. The Alter panel contains the same fields as the Define panel.

To validate the entire SCDS or any of the ACS routines, select option 4, Validate, and press Enter. To activate the CDS, select option 5, Activate, and press Enter. The CDS Name must be a valid source control data set.

Requirements and Restrictions:

- You must activate the configuration for any changes made to the SCDS to take effect.
- The SMS configuration now supports up to 32 system names, system group names, or both. When the system is running in compatibility mode (8-name mode) you can specify a maximum of eight system names, system group names or both in the SMS configuration. When the system is running in 32-name mode, you can specify a maximum of 32 system, system group names or both.

  The current system mode is defined in the IGDSMSxx member of SYS1.PARMLIB, specified when SMS is started (see “Chapter 2. Preparing for the Storage Management Subsystem” on page 9). If SMS is not active, the number of systems you can put in the SCDS is determined by the SCDS itself. If

it is an 8-name configuration, you can only put eight names in it. If it is a 32-name configuration, you can put up to 32 names in it. If it is an empty data set, it defaults to 8-name mode.

- When you access an SMS control data set (SCDS, ACDS, or COMMDS) for update which supports only eight names on a system running in 32-name mode, the data set must be converted to a new, incompatible format in order to support 32 names. You must confirm this conversion, either through the operator console or ISMF. This conversion is permanent, so you should make copies of your control data sets before converting the system mode from compatibility mode to 32-name mode.

If at any time you want to leave either the SCDS Base Define or Alter panels without saving the changed base configuration information, issue the Cancel command.

Figure 6 shows page 1 of the SCDS Base Define panel.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              SCDS BASE DEFINE                Page 1 of 2
Command ===>

SCDS NAME . : SMS.SCDS1.SCDS
SCDS Status : INVALID

To DEFINE SCDS Base, Specify:

 Description ===> SYSTEM1 CONFIGURATION
             ===>

 Default Management Class  . . TEST      (1 to 8 characters)
 Default Unit  . . . . . . . . 3380      (esoteric or generic device name)
 Default Device Geometry
   Bytes/Track . . . . . . . . 47476     (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 6. SCDS Base Define panel

Figure 7 on page 30 shows page 2 of the SCDS Base Define panel.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              SCDS BASE DEFINE                Page 2 of 2
Command ===>

SCDS NAME . : SMS.SCDS1.SCDS
SCDS Status : INVALID

Specify one of the following options . .          (1 Add, 2 Delete, 3 Rename)

  Specify System Name . . . . .          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 7. SCDS Base Define panel

As its name implies, an SCDS base configuration forms the foundation for building an SCDS.

Before you can activate an SCDS, it should be validated. The SCDS Status field tells you whether the SCDS is valid or not valid. SMS sets the status when you save an SCDS with the END command or when you select the validate option from the CDS Application Selection panel.

Figure 53 on page 146, lists conditions that cause an SCDS to be not valid. “Validating ACS Routines or an Entire SCDS” on page 143 provides more information about the validate option.

You can use the Description field to help you identify and describe the SCDS. Your description can be up to 120 characters.

The following sections explain the content of the base configuration.

Specifying the Default Management Class

For system-managed data sets that have not been assigned a management class, DFSMShsm uses the default management class for expiration, migration, and backup information. The default management class does not apply to *objects*. You specify the name of the management class in the Default Management Class field.

The default management class name is not saved in the data set catalog entry, so that you can tell the difference between data sets with assigned management classes and those without them. Specifying a default management class is optional. If a data set has no management classes and a default management class is not defined, DFSMShsm uses its own defaults for the data set.

Specifying the Default Unit

The default unit is an esoteric or generic device name, such as SYSDA or 3390, that applies to data sets that are not managed by SMS. For new data set allocations, the default unit allows users to omit the UNIT parameter from DD statements or dynamic allocation equivalent, as they can when allocating system-managed data sets. The default unit does not apply to objects.

If users do not specify the unit parameter on their DD statements or the dynamic allocation equivalent, SMS applies the default unit if the data set is non-system managed and has a disposition of either MOD (treated as NEW) or NEW. If you specify a default unit in the base configuration, make certain that it exists on the system performing the allocations. If you do not specify a default unit, end users must code the UNIT parameter to allocate data sets that are not system-managed.

Specifying the Default Device Geometry

When allocating space for a new data set on DASD, SMS converts all space requests in tracks (TRK) or cylinders (CYL) into requests for space in KB or MB. If a generic device type such as the 3380 is specified, SMS uses the device geometry for that generic device to convert tracks or cylinders into KB or MB. If an esoteric device type such as SYSDA or no UNIT is specified, SMS uses the default device geometry to convert tracks and cylinders into KB or MB. If the users in your installations specify space in tracks or cylinder units, and they specify an esoteric UNIT or no UNIT, you must specify a default device geometry prior to converting these allocations to system-managed data sets.

After SMS converts space requests to KB or MB, the space values are passed to the ACS routines. The values are later used to determine the number of tracks or cylinders to allocate for the data set. The default device geometry does not apply to objects or data sets allocated on tape.

There is only one default device geometry for the entire SMS complex. Default device geometry is an installation's definition of how much space is represented by a TRK or a CYL when an esoteric unit or no unit is specified. The device *geometry* is the track size and number of tracks per cylinder for the device.

The device geometry for 3380 is 47476 bytes per track, 15 tracks per cylinder. The device geometry for 3390 is 56664 bytes per track, 15 tracks per cylinder. It is up to each installation to decide what values to use.

Specifying the DS Separation Profile

Data set separation allows you to designate groups of data sets in which all SMS-managed data sets within a group are kept separate, on the physical control unit (PCU) level, from all the other data sets in the same group. This reduces the impact of single points of failure.

To utilize data set separation, you create a data set separation profile and specify the data set name of the profile in the base configuration. DS separation profile is an optional field that provides SMS with the data set name of the provided data set separation profile. Before you can specify this data set name, you must have created a data set separation profile. See "Appendix D. Using Data Set Separation" on page 315 to learn how to create a 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 quotes. For data set names without quotes, ISMF will add the TSO user ID prefix of the person who is defining or updating the base configuration.

Recommendation: Use a data set name that contains quotes.

The default value is **blank**, which indicates that a data set separation is not requested for the SMS complex.

Specifying Systems and System Groups in the SMS Complex

You can add, delete, or rename the systems or system groups defined to an SMS complex using page 2 of the SCDS Base Define panel, shown in Figure 7 on page 30.

A Parallel Sysplex is a collection of systems linked by closely coupled hardware facilities to process workloads. In SMS, the term *system group* is used instead of Parallel Sysplex. SMS system group name support allows you to specify a system group as a member of an SMS complex. A system group consists of system names within a Parallel Sysplex, excluding those systems that are individually specified in the SMS base configuration. A system group name can represent multiple systems. This allows SMS to support more than eight systems per SMS complex while retaining the existing configuration format of the configuration data set.

A system group name matches a Parallel Sysplex name and refers to all systems defined as part of the Parallel Sysplex that are:

- Running the same SMS configuration
- Defined in the configuration using the name of the Parallel Sysplex to which they belong (that is, system group)
- NOT defined in the configuration by their system names

For a discussion on running SMS in a Parallel Sysplex and special considerations to observe, see “Running SMS in a Parallel Sysplex Environment” on page 5.

On the SCDS Base Define panel, the System Name and Sys Group fields display the systems or system groups that you have previously defined. The system or group names appear in alphabetical order. You must add at least one system name or system group name to a base configuration before you can save it in the SCDS, and you can define up to 32 system names, system group names, or both, to the SMS complex.

Requirement: When you add, delete, or rename a system name in a base configuration, you must activate this configuration into a NULL configuration. See “Chapter 10. Activating Storage Management Subsystem Configurations” on page 171 for more information.

The system name must match the value of the SYSNAME parameter in the IEASYS*yy* member of the SYS1.PARMLIB used on that system. The system group name must match the value of the SYSPLEX parameter in the COUPLE*xx* (XCF) member of the SYS1.PARMLIB used on that system.

- To add a system name or a system group to the SCDS base configuration:

  Choose option 1, Add, and enter the system name in the System Name field or system group name in the Sys Group Name field. Press Enter to add the system.

- To delete a system or a system group from the base configuration:

  Choose option 2, Delete, and enter the system name in the System Name field or enter the system group name in the Sys Group Name field. Press Enter to delete the system or system group name from the SCDS. You are asked to confirm that you want the system or system group deleted before it is deleted.
- To rename a system or a system group in a base configuration:

  Choose option 3, Rename, enter the current name of the system in the System Name or Sys Group Name field, and enter the new name of the system in the New System/Sys Group Name field. Press Enter to rename the system or system group.

  Tip: The Rename option cannot be used to change the type of name. To change a system to a system group, or a system group to a system, you must delete the name, then add it as the desired type.

See “Recovering from a Systems Failure in the SMS Complex” on page 217 for information on recovering from a systems failure in the SMS complex.

Defining the Base Configuration for VSAM Record-Level Sharing

In order for DFSMSdfp to use the coupling facility (CF) for VSAM RLS, you must add CF cache structures to the SMS base configuration and define the cache set(s) with which they are associated.

Using Cache Structures

CF cache structures are defined to z/OS using coupling facility resource management (CFRM) policies, which determine how and where the structures are allocated. You associate these cache structures with a cache set name in the base configuration. The cache set name is also specified in a storage class definition. When a storage class associated with a data set contains a cache set name, the data set becomes eligible for VSAM record-level sharing and can be placed in a CF cache structure associated with the cache set. The system selects the best cache structure within the cache set defined for the storage class.

CF cache structures must have the same system connectivity as any storage groups that might be assigned to those cache structures. For example, if Storage Class 1 maps to cache set CS1, then the CF structures in CS1 must have the same system connectivity as the storage groups with which Storage Class 1 is associated. Connectivity of CF cache structures to all systems in the Parallel Sysplex simplifies managing and changing the configuration.

Recommendation: In a JES3 environment, be careful to define cache set names only in those SMS storage classes that are used by data sets opened for VSAM RLS processing. When you define a cache set name in a storage class, any job accessing a data set associated with that storage class is scheduled on a VSAM RLS-capable system (one where the SMSVSAM address space has been successfully initialized). If all storage classes have cache set names defined for them, then all jobs accessing SMS-managed data sets are scheduled to VSAM-RLS-capable systems. This could cause a workload imbalance between those systems and down-level systems.

See “Defining CF Cache Structures” on page 241 for more information on defining CF cache structures. See “Defining the Base Configuration” on page 27 for more information on defining the base configuration.

Defining Cache Sets

To define a cache set and specify the cache structures associated with it:

1. Select option 8 from the ISMF Primary Option Menu for Storage Administrators. ISMF displays the Control Data Set (CDS) Application Selection panel, shown in Figure 8.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                           CDS APPLICATION SELECTION
Command ===>

To Perform Control Data Set Operations, Specify:
  CDS NAME . . 'USER6.TESTCDS'
                             (1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:

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

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

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

Figure 8. CDS Application Selection for VSAM RLS

2. Specify the name of the SCDS that is to contain the base configuration for VSAM RLS in the CDS Name field. On this panel, the CDS Name field is primed with the last used SCDS.
3. Select option 7, Cache Update, and press Enter.

   ISMF displays the CF Cache Set Update panel shown in Figure 9 on page 35.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
DGTDBSA3                      CF CACHE SET UPDATE                 Page 1 of 1
Command ===>

SCDS NAME  : USER6.TESTCDS
Define/Alter/Delete CF Cache Sets:       ( 000  Cache Sets Currently Defined )

 Cache Set                      CF Cache Structure Names
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
More CF Cache Sets to Add?  . . N  (Y/N)
Use ENTER to Perform Validation; Use UP/DOWN Command to View other Pages;
Use HELP Command for Help; Use END Command to Save and Exit.
```

Figure 9. Cache Set Update Panel for VSAM RLS

4. Specify the name of a cache set. You can define up to 256 cache set names.
5. Specify the names of all CF cache structures associated with the cache set. You can specify up to eight CF cache structures for each cache set.

   Cache structures must be previously defined to z/OS in a CFRM policy.

Completing the Base Configuration

After defining base configuration attributes, you can verify their completeness and correctness by pressing Enter. If your SCDS base configuration contains any errors, the cursor moves to the field in the error and an error message appears in the short message area. Correct any errors and press Enter to verify the new contents of the SCDS base configuration.

After correcting all errors, use the END command to save the SCDS base configuration.

If at any time you want to leave either the SCDS Base Define or Alter panel without saving the changed base configuration information, issue the Cancel command.

Chapter 4. Defining Storage Groups

When managing non-system-managed DASD volumes, you view and maintain them as individual devices. If you have too many critical data sets on a volume, you have to spread the data sets across other volumes to remove I/O bottlenecks. If you are allocating data sets manually, you might not be using the volume as efficiently as you could. Some volumes containing critical data sets might be under-used. Other volumes with lower activity data sets might be overpopulated and require extra storage administrator attention to ensure that space is preserved for new data set allocations and extensions. In these and other instances, you have to work on individual volumes to solve your problems.

SMS simplifies the management of DASD, mountable optical volumes, and tape volumes by pooling them together in storage groups. This chapter describes storage groups and shows you how to define them using the ISMF Storage Group Application.

Understanding Storage Groups

Storage groups represent the physical storage managed by SMS. This storage can be collections of DASD volumes, volumes in tape libraries, volumes in optical libraries, or virtual input/output (VIO) storage. A storage group, used with storage classes, separates the logical requirements of accessing data from the physical requirements to store the data. You can use storage group attributes to specify how the system should manage the storage group. You use the storage group ACS routine to assign a new data set or object to a storage group. You can assign multiple candidate storage groups (except for objects), in which case the system chooses a specific storage group from your list. Storage group definitions are not apparent to users. Only you, as storage administrator, can define, alter, or display storage group definitions.

A storage group can be VIO, dummy, pool, object, object backup, or tape. VIO storage groups are not associated with volumes. Dummy storage groups are associated with nonexistent volumes. When defining pool storage groups, you must ensure that the actual, physical paths connecting systems to DASD volumes match the desired logical paths specified in the storage group definitions. Merely establishing a physical connection from a system to a DASD volume does not provide access. Within the storage group definition, you must specify which systems have access to which storage groups, and which storage groups have access to which DASD volumes. Likewise, merely defining a system to have access to a DASD volume does not establish a physical connection. You need to ensure that the physical connection exists.

For tape storage groups, one or more tape libraries are associated with them. Connectivity is defined at both the library level and the storage group level. If a storage group is connected to certain systems, then any libraries associated with that storage group must be connected to the same systems. Scratch volumes are added to storage groups when they are used. Private volumes can be added when they are entered in a library. Private volumes are removed from a storage group and returned to the common scratch pool when they are returned to scratch status.

For more information on how to use storage groups with objects, see “Object and Object Backup Storage Groups” on page 39.

© Copyright IBM Corp. 1986, 2002

DASD Storage Groups

When you define a pool storage group, you specify the volume serial numbers of the DASD volumes you are including in the storage group rather than their physical addresses. Each DASD volume in a storage group must contain a VSAM volume data set (VVDS) and an indexed volume table of contents (VTOC). The VVDS is automatically created after allocation of the first system-managed data set on a volume, if the VVDS is not already defined.

Two storage groups cannot share a DASD volume. You must define an entire volume to a single pool storage group. Also, a data set can only reside in one pool storage group. A data set can span volumes within a single pool storage group, but it cannot span volumes belonging to several pool storage groups. For VSAM data sets, the entire sphere (base cluster and all alternate indexes) must be in the same storage group.

Recommendation: Define pool storage groups so that they only contain devices of the same geometry. The device *geometry* is the track size and number of tracks per cylinder for the device.

3390 devices in 3380 track compatibility mode are geometrically the same as 3380 devices, and the access methods see them as 3380 devices. Therefore, you can combine these devices in a single storage group.

Requirement: To enable extend processing in the extend storage group, ensure that the extend storage group contains devices with the same geometry as the initial storage group.

Although you can separate devices according to geometry, you do not need to separate them according to capacity. For example, you can combine all models of the 3390 into a single storage group. The only effect the different capacities has is on volume thresholds. See *z/OS DFSMS: Implementing System-Managed Storage* and *MVS/ESA SML: Managing Storage Groups* for information on selecting appropriate threshold levels.

Devices of the same geometry can have different performance characteristics. These devices coexist in the same storage group, and enhanced volume selection for SMS manages data set placement accordingly. Even devices with vastly different performance characteristics can reside in the same storage group.

For striped data sets through sequential access, the effective data rate is limited by the slowest stripe. If a storage group contains volumes with widely different data delivery capabilities, such as 3990 Model 6 and ESS, the effective data rate for striped data sets through sequential accesses is gated by the stripes on 3990 Model 6.

DASD Volume Status for Data Sets

To prepare new DASD volumes for SMS storage groups, use the Device Support Facilities (ICKDSF) INIT command to assign new DASD volume serial numbers and allocate an indexed VTOC. You can use the STORAGEGROUP (STGR) keyword of the INIT command to make a DASD volume available for allocation of new system-managed data sets.

You can use INIT from the ISMF Volume Application and the Storage Group (through LISTVOL) Application. If you go to the Volume Application directly from the Primary Option Menu, you initialize the DASD volume as a non-system-managed

volume. If, instead, you work from a DASD volume list generated within the Storage Group Application, you initialize the DASD volume as a system-managed volume. When new volumes are added to a storage group in the active configuration, we recommend explicitly allocating small temporary data sets to each new volume to update the SMS volume space statistics.

As you bring DASD volumes under the control of SMS, you need to keep track of those that are ready to be system-managed. The physical DASD volume status can be one of the following: converted, initial, non-SMS, or unknown. To display this information, on the Storage Group Application panel, enter the LISTVOL command against pool storage groups.

Convert status indicates that the DASD volume is converted and is fully available for SMS control. To be converted, all data sets on the DASD volume must have an associated storage class, and all permanent data sets on the DASD volume must be cataloged in an integrated catalog facility (ICF) catalog. The DASD volume must have a VTOC index.

Initial status indicates that an attempt to convert the DASD volume has been made, but the DASD volume contains data sets that fail to satisfy the requirements of a converted volume. In the initial state, no new allocations can be made on the DASD volume. You cannot extend data sets to additional DASD volumes. The initial status allows DFSMSdss™ to process the DASD volume without having to issue a RESERVE for the entire duration of the conversion process. Also, to be eligible for conversion, data sets and the DASD volume must have an organization that is supported by SMS. For more information, refer to *z/OS DFSMS: Implementing System-Managed Storage*.

Object and Object Backup Storage Groups

An object storage group is a storage group that defines the physical storage used for objects. For objects, the storage group allows you to define an object storage hierarchy. An object storage hierarchy consists of an object directory and DB2 object storage tables, and may also contain one or more of the following options:

- Library or shelf resident optical volumes
- Library or shelf resident tape volumes

You can define an object storage group that does not include an optical library. If an object storage group does not include an optical library, the objects reside on DASD volumes or tape volumes and do not get moved to optical volumes.

An object can move up and down the hierarchy within a single storage group but cannot move outside of that group. Objects move within the storage hierarchy depending initially on management class and storage class criteria for the object collection, and subsequently on ACS routines.

You can define two types of storage groups for the object access method (OAM): object and object backup. An object storage group defines an object storage hierarchy. The object backup storage groups define the optical libraries or tape units that are used for backing up objects. An object or object backup storage group can be enabled by and connected to more than one system in a Parallel Sysplex. There is no limit on the number of object or object backup storage groups defined to each system in the SMS complex. Up to two object backup storage groups can be associated with each object storage group.

Two timing attributes are associated with object storage groups: Cycle Start Time and Cycle End Time. These attributes define a window during which the OAM Storage Management Component (OSMC) can automatically start its storage management processing for the storage group. The window occurs once each day. An object backup storage group has no timing attributes.

You can also specify an OSMC processing system name, which identifies which OAM system in the sysplex performs OSMC processing for this object storage group. This prevents multiple systems from trying to process the same storage group at the same time, especially when the Cycle Start Time is used for automatically starting storage group processing.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information about managing objects.

Defining Real and Pseudo Optical Libraries in Object Storage Groups

When defining an object storage group, you can specify up to eight real optical libraries or up to eight pseudo optical libraries. You cannot mix real and pseudo libraries in a storage group definition. If the object's storage class indicates that the object should reside on optical, OAM stores the object using any optical drive associated with any of the optical libraries listed in the object storage group definition, and any eligible optical volume that is accessible by the drive selected.

The object or object backup storage groups can share optical libraries. A single optical library can contain optical volumes belonging to several object or object backup storage groups. There can be multiple object storage groups per system.

Defining Optical Libraries in Object Backup Storage Groups

When defining an object backup storage group, you can specify up to eight real optical libraries or up to eight pseudo optical libraries.

OAM stores the backup copy or copies of the object using any optical drive and optical volume that is associated with any one of the optical libraries listed in the object backup storage group definition.

If no optical libraries are defined to the object backup storage group, OAM stores the backup copy or copies of the object using the tape unit name specified for the object backup storage group in the CBROAMxx member of PARMLIB.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information about defining optical libraries in object and object backup storage groups.

Writing Objects

The storage class for an object determines where an object is written (DASD, optical, or tape). Objects can be written to tape using the media type and tape unit name defined in the SETOAM statements in the CBROAM*xx* member of PARMLIB.

When the storage class indicates the object should reside on optical, objects can be written using any drive in any library associated with the object storage group, using any volume assigned to that storage group, as long as there is enough space to satisfy the request and the volume is write-compatible with the drive. If no group volumes are available, the request is satisfied using a scratch volume residing in a library associated with the object storage group.

When the storage class indicates the object should reside on tape, the object can be written using any volume assigned to that storage group, using the unit name in the tape volume record for allocation of a tape device. If no group volumes are available, the request is satisfied using a scratch volume allocated using the tape unit name specified for that storage group in the SETOAM command in the CBROAM*xx* member of PARMLIB.

Both object and object backup storage groups have three attributes that are concerned with writing objects:

- The drive startup threshold
- The volume full threshold or tape full threshold
- The mark volume full on first write failure option

The attributes are specified in the CBROAM*xx* member of PARMLIB.

The *drive startup threshold* is the maximum number of write requests (for optical) or maximum KB of object data (for tape) that can be waiting for each drive that is currently processing write requests for the storage group. If the threshold is specified and then exceeded, an attempt is made to start another drive to process write requests to the storage group.

When the number of free KB on a volume falls below the *volume full threshold* specified for the storage group, the volume is marked full. Similarly, when the number of free KB on a volume falls below the *tape full threshold*, the tape volume is marked full. If you specified *mark volume full on first write failure*, the volume is marked as full the first time an attempt to write an object on the volume fails because not enough space remains on the volume.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information about writing objects.

OAM Collection Names

OAM collection names provide another level of grouping called a *collection*. A collection is a group of objects that share a common storage group and have the same default initial storage class and management class attributes. All objects within a collection must have unique names. However, you can have two objects with identical names if the objects belong to separate collections.

OAM collections are cataloged in integrated catalog facility (ICF) catalogs using OAM collection names. An individual object does not have an entry in the catalog.

Each object is a member of a collection and each collection is part of one storage group. A collection does *not* span storage groups. OAM identifies an object by its collection name and its object name. An object is described by an entry in a DB2 object directory. A collection is described by an MVS collection name catalog entry and a corresponding OAM collection identifier table entry.

OAM provides two primary interfaces for processing objects: the application program interface (OSREQ) and the OAM Storage Management Component (OSMC). Both address objects using the object's collection name and object name including those cases where the object must be identified to the ACS routines for the STORE (OSREQ), CHANGE (OSREQ), and CTRANS (OSMC class transition) environments. Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Application Programmer's Reference* for more information.

Tape Storage Groups

The Storage Group Application supports a tape storage group type for tape libraries. The LISTVOL line operator on the Storage Group List supports the mountable tape volume list.

The storage administrator can specify tape as the storage group type and can then define and alter tape storage groups and their relationships to the systems in the SMS complex. These relationships are NOTCON, ENABLE, DISALL, DISNEW, QUIALL, or QUINEW. See “Defining System Access to Pool and VIO Storage Groups” on page 48 for an explanation of these relationships.

A tape storage group is a collection of tape cartridges and is associated with one to eight tape libraries. A scratch volume is added to a storage group on use. A private volume can be added to a storage group when it is entered in a library. You can direct allocations to a local or remote library or to a specific library by assigning the appropriate storage group in the storage group ACS routine.

Related Reading:

- If you use the DFSMSrmm storage group support for managing scratch tape pools, you must define the storage groups as tape storage groups. See *z/OS DFSMSrmm Implementation and Customization Guide* for further information.
- For more information about tape storage groups and tape libraries, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Library Definition

Tape libraries are supported using MVS/Hardware Configuration Dialog (MVS/HCD) generated input/output definition file (IODF). If MVS/HCD is not currently being used, the installation must convert from MVS configuration program (MVSCP) to MVS/HCD. If the LIBRARY-ID/LIBPORT-ID fields on the ‘Define Device Parameters / Features’ HCD panel are utilized and correctly filled in, only an Activate of the IODF followed by a vary online of the drives will be required to use the tape library and exploit the tape library support. If the LIBRARY-ID/LIBPORT-ID fields on the ‘Define Device Parameters / Features’ HCD panel are not utilized, an IPL is required to use the library and exploit the tape library support.

SMS

For SMS support of the tape library, an SCDS of the proper level must be activated. This SCDS must contain appropriate library definitions and tape storage group definitions. Information that the storage administrator must provide specifically for an Automated Tape Library Dataserver (ATLDS) or manual tape library (MTL) is listed below. See *z/OS DFSMS: Implementing System-Managed Storage* for more information about MTL and ATLDS.

- Data compaction, media type and recording technology information can be included in data classes that are used in allocation of devices in a tape library.
- New storage groups of type TAPE must be defined. The storage groups are involved in the allocation of new data sets to volumes within a tape library.
- A tape library must be defined in the SCDS.
- Default data class defined to the library.
- ACS routines must specify a storage class and the storage group selected must be associated with a tape library.

Planning Storage Groups for Data Sets

Ideally, you would have only one storage group containing all of your data sets, and the system would manage everything. Realistically, you have to account for a variety of data sets, such as databases, large data sets, temporary data sets, and tape data sets.

Before you actually define your storage groups, you should gather the following information:

- Existing I/O hardware configuration
- Projected hardware requirements
- Estimated general requirements of user groups
- Anticipated security requirements
- Backup and recovery requirements
- Data set sizes
- Required system access, based on both shared data requirements and the needs of user groups
- The number of systems to which the storage group is connected

To begin your planning, define one primary storage group and identify all the data sets that do not fit into this general storage group category. Keep in mind your storage class and management class requirements.

For data sets that require continuous access (continuous for the storage class availability attribute) the storage group must contain sufficient volumes that do not interrupt data availability for a single device failure (for example, dual copy or array DASD).

The highest performance is generally obtainable through use of cache at the subsystem level, the device level, or both. Sufficient volumes with cache access should be available in a storage group for data sets that require high performance. Data sets that have primarily write access (i.e., write BIAS in the storage class) benefit most from a DASD fast write capability.

For data sets requiring concurrent copy, ensure that sufficient volumes in the storage group are attached through 3990 Storage Controls with Extended Platform, or through an IBM RAMAC Virtual Array.

For striped data sets, ensure that there are a sufficient number of separate paths to DASD volumes in the storage group to allow each stripe to be accessible through a different path. The maximum number of stripes for one data set is 16. (A stripe is the portion of a striped data set that resides on one volume.)

Figure 10 shows the formula that is used to derive the optimum number of stripes that are allocated to the data set:

```
Sustained data rate / Device transfer rate = Optimum number of stripes
```

Figure 10. Formula for Deriving Optimum Number of Stripes

where:

- Sustained data rate is the target throughput rate, specified in the storage class definition

- Device transfer rate is based on the published characteristics of the device

Ideally, the number of stripes should be on different paths, however, if this is not possible, SMS allocates to a smaller number of stripes.

If a data set has a management class that specifies automatic backup or migration, you must direct the data set to a storage group that is eligible to be processed for automatic backup or migration. You need to specify through ISMF the system or system group that is to perform the backup or migration only if that particular system or system group should do the processing. Otherwise, do not specify a particular system or system group to do the processing. It is done automatically. Management class attributes do not apply to tape data sets, so only assign data sets to tape which do not require management class services.

Planning Storage Groups for OAM Object Collections

For each major application that processes objects, one or more object collections must be defined to provide for the storing, cataloging, and retrieval of objects used by that application. Each object collection must be assigned to a storage group by the storage group ACS routine. Multiple collections can be assigned to the same storage group.

A storage group for OAM object collections provides a storage hierarchy containing a DB2 table for the object directory for objects in the storage group, DB2 tables for storage of objects on DASD, optical volumes for the storage of objects on optical media, and tape volumes for the storage of objects on tape media.

OAM uses the following method, using the definition of the storage class currently assigned to the object, to determine whether to store objects on DASD, tape or optical:

- If the Initial Access Response Seconds (IARS) attribute specified in the storage class equals zero, objects are stored on DASD.
- If the IARS is greater than 0 and the Sustained Data Rate (SDR) is less than 3, objects are stored on optical.
- If the IARS is greater than 0 and the SDR is greater than or equal to 3, objects are stored on tape.

Before you actually define your storage groups, you should gather the following information:

- Existing I/O hardware configuration
- Projected hardware requirements
- Estimated general requirements of user groups
- Anticipated security requirements
- Backup and recovery requirements
- Object sizes
- Required system access, based on both shared data requirements and the needs of user groups
- The number of systems to which the storage group is connected

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information on planning for objects.

Defining Storage Group Attributes

You can use ISMF to define your storage groups by selecting option 6, Storage Group, from the ISMF Primary Option Menu for storage administrators. Figure 11 shows the Storage Group Application Selection panel.

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

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . . . 'SMS.SCDS1.SCDS'
                                (1 to 44 character data set name or 'Active' )
  Storage Group Name  . . VI01               (For Storage Group List, fully or
                                             partially specified or * for all)
  Storage Group Type  . . VIO                (VIO, POOL, DUMMY, OBJECT, OBJECT
                                             BACKUP, or TAPE)

Select one of the following options  :
  2  1. List    - Generate a list of Storage Groups
     2. Define  - Define a Storage Group
     3. Alter   - Alter a Storage Group
     4. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 11. Defining a Storage Group

To define a storage group, you must specify a control data set name, storage group name, and storage group type on the panel and select option 2, Define. The control data set name must be the name of an SCDS. ISMF primes the field with the name last used within ISMF. (The default is 'active', which represents the currently active configuration, but you cannot define or alter the storage groups for the 'active' configuration.)

In the Storage Group Name field, you must specify the name of the storage group that you are defining. ISMF primes the field with the name last used within ISMF. The default is an asterisk, *, which represents all storage groups in the specified control data set name.

In the Storage Group Type field, you must specify one of the following types:

VIO Virtual I/O (VIO) storage groups are used to allocate data sets to VIO, which simulates the activity of a DASD volume. VIO storage groups do not contain any actual DASD volumes. You can put temporary data sets in VIO storage groups.

Pool Pool storage groups contain the volume serial numbers of system-managed DASD volumes. Pool storage groups can be used for both temporary and permanent data sets.

Dummy Dummy storage groups contain the volume serial numbers of DASD volumes that no longer reside on the system but that you want to treat as SMS DASD volumes. Using dummy storage groups allows existing JCL that explicitly references the DASD volumes in

VOL=SER statements to work. If end users specify a VOL=SER in their JCL, and that volume serial number is in a dummy storage group list, then SMS issues a catalog request to find the desired data set rather than using the volume serial number.

Volumes in dummy storage groups cannot be used when performing volume allocations. For example, the following DD statement, where DUMMY1 is a volume in a dummy storage group, does not work:

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

A dummy storage group should not contain the volume serial number of a DASD volume that exists in the system. If the DASD volume exists in the system and the data set is system-managed, no JCL errors occur but the job fails during allocation. If the DASD volume exists in the system and the data set is not system-managed, then the resulting errors depend on the type of data set. For uncataloged data sets, either the data set cannot be found or the wrong data set with the same name is found. For cataloged data sets, the job fails during allocation.

Object
Object storage groups identify an object storage hierarchy.

The object storage groups can be defined with no optical libraries and have the backup copies of objects written to tape by specifying SETOAM statements in the CBROAM*xx* member of PARMLIB.

Object Backup
Object backup storage groups define the groups that are to be used to contain backup data and can specify one or more optical libraries that contain backup copies of objects.

The object backup storage groups can be defined with no optical libraries and have the backup copies of objects written to tape by specifying SETOAM statements in the CBROAM*xx* member of PARMLIB.

Tape
Tape storage groups identify storage groups to maintain system-managed tape volumes.

After specifying a storage group type, select option 2, Define, and press Enter to reach the appropriate Storage Group Define panel.

The remainder of this chapter provides the panel sequences in the Storage Group Application and explains the attribute values for each type of storage group. You can leave any of the Storage Group Define panels at any time without saving the storage group by issuing the Cancel command.

Defining a VIO Storage Group

Figure 12 on page 47 shows the VIO Storage Group Define panel.

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

SCDS Name  . . . . : SMS.SCDS1.SCDS
Storage Group Name : VIO1

To DEFINE Storage Group, Specify:
 Description ==>
             ==>
 VIO Maxsize . .                   (8 to 2000000)
 VIO Unit  . . .                   (DEVICE TYPE)

 System/Sys                   System/Sys
 Group Name      Status       Group Name      Status
 ----------      ------       ----------      ------
 SYS1       ===> ENABLE                  ===>          * SYS GROUP = sysplex
            ===>                         ===>            minus systems in the
            ===>                         ===>            sysplex explicitly
            ===>                         ===>            defined in the SCDS
            ===>                         ===>
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 12. Defining VIO Storage Group Attributes

The SCDS Name and Storage Group Name fields are output fields that contain the SCDS and storage group names you specified in the Storage Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the VIO storage group.

To define a VIO storage group, you must specify a data set maximum size for VIO and a device type. You must also assign a storage group status for each system and system group.

After specifying the attribute values and system status relationships, issue the END command. END saves the newly defined VIO storage group and returns you to the Storage Group Application Selection panel.

VIO Maxsize

VIO Maxsize is a required value that represents the maximum size, in KB, of data sets to be allocated to VIO. If a data set exceeds the maximum size, then the allocation fails. However, if the storage group ACS routine assigns both a VIO storage group and a pool storage group to the list of candidate storage groups for the data set, and the data set exceeds the maximum size, then the data set is allocated to the pool storage group.

Caution: z/OS supports two space-related parameters called MXIG and ALX. Use these parameters with extreme caution for VIO as well as non-VIO allocations. You can specify them as subparameters of the JCL SPACE keyword. You can also specify them to the system as defaults in the allocation parmlib member ALLOCxx, in which case they get applied for VIO and dynamic allocation requests if space parameters are not specified. In these cases instead of allocating the requested space quantity the allocated space is based on MXIG or ALX values. See *z/OS DFSMS: Using Data Sets* for details on the actual amount of space allocated. For VIO allocations this can result in impacting paging space and continued operation of the system if all of the allocated space is actually used. These parameters can be overridden by the dynamic allocation installation exit. The value passed to the SMS

ACS routines for the read-only variables &SIZE and &MAXSIZE are based on the user-specified space quantity and are not based on MXIG or ALX.

VIO Unit

VIO Unit is a required field that represents the generic DASD device type (for example, 3380 or 3390) that this storage group simulates.

Defining System Access to Pool and VIO Storage Groups

If you want a system to access a pool storage group, you must ensure that devices on which the groups reside are physically connected to the system.

No devices are associated with a VIO storage group. However, the SMS status associated with the VIO storage group determines if this storage group can be used by each system in the complex. The SMS statuses, described below, show the relationships between each system in the SMS complex and a given VIO or pool storage group.

DISALL Disable All prevents the system from allocating or accessing data sets in the storage group.

DISNEW Disable New prevents the system from allocating new data sets (and DISP=MOD data sets that do not currently exist) in the storage group. DISNEW prevents data sets from extending to a volume whose volume or storage group status is DISNEW.

ENABLE The system can allocate and access data sets in the storage group. ENABLE is the default relationship between a system and a storage group.

NOTCON Not Connected indicates that the storage group is defined but not accessible to the system. It resembles DISALL, except you cannot dynamically change the NOTCON system status. If you want a storage group to have access to NOTCON volumes, physically connect them to the system, define the connection by changing the system status in the ISMF Storage Group Application, and activate the configuration that defines the connection.

QUIALL Quiesce All prevents the system from scheduling jobs that allocate or access data sets in the storage group. This state only affects JES3 systems, that support job scheduling.

When the job is executing, QUIALL has the same effect in a JES2 or JES3 environment. In an JES2 environment, the volumes become a secondary volume selection candidate through QUIALL. *Secondary volume selection candidacy* means that these volumes are still available for allocation, but less preferred than other volumes.

For more information about primary and secondary volume selection candidacy, see “Conventional Volume Selection” on page 103.

QUINEW Quiesce New prevents the system from scheduling jobs that allocate new data sets (and DISP=MOD data sets that do not currently exist) in the storage group.

If the system uses JES3 to schedule jobs, you can use QUINEW only as long as other volumes are available. Before scheduling a job, JES3 verifies that all resources are available, unlike JES2 which schedules a job even if all the needed resources are not available. This can lead to a contention for resources, since a job

running under JES2 can hold some resources while it waits for others to become available. Under JES3, if any of the candidate volumes are available, JES3 schedules the job and SMS selects the volume. Under both JES2 and JES3, SMS selects QUINEW volumes only as a last resort.

For more information about primary and secondary volume selection candidacy, see “Conventional Volume Selection” on page 103.

Defining a Dummy Storage Group

Figure 13 shows the Dummy Storage Group Define panel.

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

SCDS Name . . . . . : SMS.SCDS1.SCDS
Storage Group Name  : DUMMY1

To DEFINE Storage Group, Specify:

 Description ==>
             ==>









Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 13. Defining Dummy Storage Group Attributes

SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel.

Description is an optional field of 120 characters where you can describe the dummy storage group. Issue the END command to save the newly defined dummy storage group and return to the Storage Group Application Selection panel.

Initially, the storage group contains no volume serial numbers. To add volume serial numbers to the storage group, select option 4, Volume, from the Storage Group Application Selection panel shown in Figure 11 on page 45. ISMF primes the other fields on the panel with the most recently specified values. (You do not need to specify a Storage Group Type.) Then, press Enter to define volume serial numbers to the storage group in the Storage Group Volume Selection panel shown in Figure 14 on page 50.

On this panel, you specify the volume serial numbers that you want to belong to the dummy storage group.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                          STORAGE GROUP VOLUME SELECTION
Command ===>

CDS Name  . . . . . : SMS.SCDS1.SCDS
Storage Group Name  : DUMMY1
Storage Group Type  : DUMMY

Select One of the following Options:
  2  1. Display      - Display SMS Volume Statuses (Pool only)
     2. Define       - Add Volumes to Volume Serial Number List
     3. Alter        - Alter Volume Statuses   (Pool only)
     4. Delete       - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
      Prefix   From     To    Suffix  Hex

      ______  ______  ______  _____    _
 ===> SYS     001     077                     ('X' in HEX field allows
 ===> DFPIP1                                   FROM - TO range to include
 ===> SYS     25      30      A        X      hex values A through F.)
 ===>         001     010     VOL
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 14. Defining Dummy Storage Group Volume Serial Numbers

To add DASD volumes to the dummy storage group, select option 2, Define, and specify the volume serial numbers in the Specify a Single Volume (in Prefix) or Range of Volumes field.

You can specify a single volume serial number by typing the number under Prefix. The value can be from one to six characters, but it must be a fully specified volume serial number. You can specify a range of volume serial numbers by typing the prefix or suffix that is common to a set of volumes under Prefix or Suffix, the low individual volume number under From, and the high individual volume number under To.

You can specify an X under Hex to include hexadecimal values in your range. For example, in Figure 14 volume numbers SYS001 through SYS077 are specified on the first volume specification line. Volume number DFPIP1 is specified on the second line. Volume numbers SYS25A through SYS30A, including SYS2AA through SYS2FA, are specified on the third line. Volume numbers 001VOL through 010VOL are specified on the fourth line. You can add as many as a hundred volume serial numbers at a time to a pool storage group this way.

After you specify volume serial numbers, press Enter to add the DASD volumes to the dummy storage group. Define any additional volume serial numbers, then use END to return to the Storage Group Application Selection panel. If any volumes cannot be added, a list of the volumes that could not be added is displayed.

When adding a new volume serial number (volser), you must make sure that all volsers defined in the volume list are unique. ISMF verification only verifies that all volsers are unique within the same dummy or pool storage group. No attempts are made to verify the device types of each volume. Therefore, if a library volser is duplicated in a dummy or pool storage group, this error is not detected until a request is issued to mount this volser.

Defining a Pool Storage Group

Figure 15 shows the Pool Storage Group Define panel. The SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel.

Description is an optional field of 120 characters where you can describe the pool storage group.

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

SCDS Name . . . . . : SMS.SCDS1.SCDS
Storage Group Name  : POOL1
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 . . . . . . . .

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

 Allocation/migration Threshold: High . . 85  (1-99)      Low  . .     (0-99)
 Guaranteed Backup Frequency  . . . . . .          (1 to 9999 or NOLIMIT)

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

Figure 15. Defining Pool Storage Group Attributes

The storage group and management class are interrelated. The storage group Auto Migrate and Auto Backup parameters specify whether the volumes in this storage group are eligible to be processed automatically. The management class, assigned to the data sets residing on the volumes, determines whether and how to process the data sets on the volume. In contrast, if you set Auto Migrate or Auto Backup to **N**, no, in the storage group, the volumes in the storage group are not processed and data sets residing in the storage group are not migrated or backed up.

You can specifiy the following attributes on the Pool Storage Group Define panel:

Auto Migrate specifies whether the DASD volumes in this storage group are eligible for automatic space management processing. Auto Migrate is a required field, which ISMF primes with the value **Y**, yes.

Y specifies that DFSMShsm is to perform automatic space management processing if processing windows are defined to DFSMShsm. Data sets are eligible for primary space management. If the Setsys Intervalmigration attribute has been specified in DFSMShsm, the data sets are also eligible for interval migration.

N specifies that data sets are not eligible for automatic migration.

I specifies that data sets are eligible for primary space management and specifies that DFSMShsm is to perform automatic interval

migration independent of the DFSMShsm Setsys Intervalmigration option. **I** also causes DFSMShsm to perform the same functions as if **Y** had been specified. Interval migration is performed on only those volumes whose occupancy is at or above their high threshold. The most frequent DFSMShsm can do interval migration is hourly. If you select **I** with a low threshold value of 0, DFSMShsm migrates all the eligible data sets in the selected storage group. **I** is most useful for storage groups used with tape mount management.

P specifies that data sets are eligible for primary space management but interval migration is not performed even if Setsys Intervalmigration is specified.

If for some reason (such as a system outage) system-managed temporary data sets are not deleted at the end of a job, DFSMShsm attempts to delete them during primary space management. In order for DFSMShsm to delete these data sets, be sure to specify Auto Migrate as **Y** or **I**, or Auto Backup as **Y**.

Auto Backup specifies whether the DASD volumes in this storage group are eligible for automatic backup processing. Auto Backup is a required field, which ISMF primes with the value **Y**, yes. **Y** specifies that DFSMShsm backs up the data sets on the volume according to management class requirements.

Auto Dump In Auto Dump, you specify whether you want to automatically dump all the DASD volumes in this storage group. It is an optional field, which ISMF primes with the value **N**, no. If you specify a value of **Y**, yes, DFSMShsm automatically dumps the volumes.

Migrate, Backup, and Dump Sys/Sys Group Name

ISMF no longer verifies that specified system or system group names are defined to the base configuration. You can specify either a system or a system group name in these fields but a specific system specified might not be defined in the configuration since it might be defined as part of a system group. Therefore, if these fields are not blank, you must take special care to ensure that the values are correct. If the values are incorrect, it could result in the expected DFSMShsm operation not occurring.

DFSMShsm has the capability of processing each storage group for automatic space management, data availability management, and automatic dump processing. A system is eligible to perform the processing when any of the following are conditions are met:

- The name is blank, meaning any system can perform it
- The name specified is the name of the specific host system
- The name specified is the name of the system group to which the system belongs and the system is not individually defined by its system name in the configuration.

DFSMShsm ignores storage groups for which you specify a different system name, and does not process DASD volumes that have already been processed. Do not specify a Sys/Sys Group Name unless processing of the storage group for the function must be performed only on that one host because that limits the capabilities of DFSMShsm to perform the request.

The same rules apply to Backup Sys/Sys Group Name for data availability management processing, and to Dump Sys/Sys Group Name for automatic dump processing. All three Sys/Sys Group Name fields are optional and primed with blanks.

Overflow
specifies whether this pool storage group is designated as an overflow storage group to handle periods of high demand for initial primary space allocations. **Y**, yes, specifies that the storage group is an overflow storage group.

Overflow storage groups are utilized when the enabled (non-overflow) storage groups are above high threshold.

Overflow is a required field, which ISMF primes with the value **N**, no.

During initial allocation, volumes in an overflow storage group are less preferred than volumes in an enabled storage group but more preferred than volumes in a quiesced storage group. Therefore, overflow volumes are not placed on the primary volume list but can be placed on the secondary volume list. When an overflow storage group contains more volumes than a buffer storage group, specified volume counts might result in overflow volumes being preferred over volumes in the buffer storage group.

An overflow storage group may also be specified as an extend storage group.

Volumes residing in overflow storage groups are preferred over quiesced volumes and storage groups. If you quiesce an overflow storage group or volume then the quiesced volumes are preferred over quiesced overflow volumes.

Recommendation: Because overflow storage group status is not recognized on systems prior to z/OS R1V3, set the overflow storage group status to ″quiesced″ on lower-level systems. This preserves overflow storage groups for peak usage.

Extend SG Name
specifies the name of another pool storage group to which data sets from the primary storage group can be extended when there is an insufficient amount of storage on the primary storage group. A primary storage group is the storage group in which the initial allocation resides.

Extend SG Name is an optional field of one to eight alphanumeric or national characters, or both, the first of which must be alphabetic. ISMF primes the field with the default value “blank.”

You can define only one extend storage group to each storage group. However, you can define the same extend storage group to more than one primary storage group. Also, you can define two storage groups as extend storage groups of each other.

An extend storage group may also be specified as an overflow storage group.

Extend storage groups will not be used for initial allocation unless they are specified in the ACS routines. All storage groups that are listed in the ACS routines are candidates for initial allocation. Extend storage group attributes are not referenced during initial allocation.

Example: You can define storage group (SG) 3 as an extend of SG 2, and define SG 2 as an extend of SG 1. If SMS selects SG 1 as the primary storage group, the data set can extend only to SG 2. If SMS selects SG 2 as the primary storage group, the data set can extend only to SG 3.

For more information about data set extend in the volume selection process, see “SMS Volume Selection for Data Set Allocation” on page 102.

Requirement: When you specify an extend storage group, you must ensure that connectivity across the SMSplex is the same for both the primary storage group and the extend storage group.

Dump class
specifies the name of a dump class that is defined in DFSMShsm. There are five Dump Class fields, so up to five unique dump class names may be specified.

When DFSMShsm dumps DASD volumes that belong to the storage group, it directs their contents to the dump classes. To use dump classes, you must define their names and parameters using DFSMShsm. Then on this panel, you can identify the dump class names to SMS.

Dump class is an optional field that is primed with the default value ″blank.″

Migration and Allocation Thresholds
specifies an upper and lower space limit for the DASD volumes in a pool storage group.

SMS tries to stay within these values by looking at a data set’s primary space allocation before assigning it to a given DASD volume. For example, the SMS volume selection function attempts to prevent allocation of a data set to a given DASD volume if that allocation causes the volume’s high threshold to be exceeded. In addition, this high threshold value is used by DFSMShsm to determine whether data sets should be migrated off a DASD volume in the storage group. The low threshold value is used as the threshold goal in reducing the amount of space occupied on a DASD volume in the storage group during interval migration or daily space management. The low threshold value must be less than or equal to the high threshold value.

Both numbers are percentages of the total space on the DASD volume. If you specify **Y** for Auto Migrate, then you must specify both a high and low threshold. If you specify **Y**, the low threshold limit is 1. If you specify **I** for automatic interval migration, you can specify a low threshold value of 0 to migrate all the data sets in the selected storage group. The hourly migration trigger for storage groups with a value of AM=I is the occupancy at or above the average of high and low thresholds. Since storage groups used for tape mount management tend to fill up several times a day, allowing interval migration for these storage groups allows DFSMShsm to better keep up with the demand.

ISMF requires that you enter a high threshold value when you specify **N** for Auto Migrate, when defining a pool storage group. Since SMS needs the value for allocation purposes, the High Threshold field is a required field with a primed value of 85.

For more information on specifying allocation and migration thresholds, see *z/OS DFSMShsm Storage Administration Reference*.

Guaranteed Backup Frequency
specifies the number of days within the last backup period in which the backup process should have a copy of each of the data sets within the applicable storage group. If Auto Backup is specified as **Y**, this attribute is required; otherwise it is optional.

You specify the maximum number of days that can elapse between backups. You can specify from 1 to 9999 days or you can specify NOLIMIT. If you specify NOLIMIT, then data sets in the storage group are backed up according to management class specifications. There is no default.

Defining Pool Storage Group Status

In the SMS Storage Group Status Define panel, indicate whether you want to change the status of the pool storage group with respect to a given system in the SMS complex. Initially, all of the status fields are ENABLE. Specify **Y**, yes, to change a status. Accept the default **N**, no, to leave all the status fields as ENABLE.

When you have completed all your entries in the Pool Storage Group Define panel, press Enter. If you specified **Y** for the Define SMS Storage Group Status attribute, you get the SMS Storage Group Status Define panel shown in Figure 16 , where you can define the status of the storage group to each system in the SMS complex.

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

SCDS Name  . . . . : SMS.SCDS1.SCDS
Storage Group Name : POOL1
Storage Group Type : POOL
To DEFINE Storage Group System/                            ( Possible SMS SG
Sys Group Status, Specify:                               Status for each:
                                                         - Pool SG Type
 System/Sys      SMS SG    System/Sys      SMS SG          NOTCON, ENABLE
 Group Name      Status    Group Name      Status          DISALL, DISNEW
 ----------      ------    ----------      ------          QUIALL, QUINEW
 SYSTEM1    ===> ENABLE    SYSTEM2    ===> ENABLE       - Tape SG Type
 SYSTEM3    ===> ENABLE    SYSTEM4    ===> ENABLE         NOTCON, ENABLE,
 SYSTEM5    ===> ENABLE    SYSTEM6    ===> NOTCON         DISALL, DISNEW )
 *SYSPLX1   ===> ENABLE               ===>
            ===>                      ===>              * SYS GROUP = sysplex
            ===>                      ===>                minus Systems in the
            ===>                      ===>                Sysplex explicitly
            ===>                      ===>                defined in the SCDS
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 16. Defining Pool Storage Group System Status

The system or system group names you defined in the base configuration are listed under System/Sys Group Name. SMS SG Status lists the relationship between the storage group and each system in the SMS complex. You can specify the statuses, which are explained in “Defining System Access to Pool and VIO Storage Groups” on page 48.

After establishing the relationships between your storage group and the systems in the SMS complex, you need to define DASD volumes to the storage group. From the SMS Storage Group Status Define panel, enter the END command to save your values and return to the Pool Storage Group Define panel. On this panel, enter the END command to save the pool storage group. This returns you to the Storage Group Application Selection panel.

Adding Volumes And Defining the SMS Volume Status

On the Storage Group Application Selection Panel (Figure 11 on page 45) select option 4, Volume, and press Enter to view the Storage Group Volume Selection panel which appears in Figure 14 on page 50.

Select option 2, Define, and specify the volume serial numbers of the DASD volumes that you want to add to the pool storage group. Each time you press enter, you see the SMS Volume Status Define panel shown in Figure 17.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                          SMS VOLUME STATUS DEFINE             Page 1 of 2
Command ===>

SCDS Name . . . . . . : SMS.SCDS1.SCDS
Storage Group Name  . : POOL1
Volume Serial Numbers : 010000 - 010010

To DEFINE SMS Volume Status, Specify:

 System/Sys      SMS Vol     System/Sys      SMS Vol    ( Possible SMS Vol
 Group Name      Status      Group Name      Status       Status for each:
 ----------      -------     ----------      -------      NOTCON, ENABLE,
 SYSTEM1    ===> ENABLE      SYSTEM2    ===> ENABLE       DISALL, DISNEW,
 SYSTEM3    ===> ENABLE      SYSTEM4    ===> ENABLE       QUIALL, QUINEW )
 SYSTEM5    ===> ENABLE      SYSTEM6    ===> ENABLE
 *SYSPLX1   ===> ENABLE                 ===>            * SYS GROUP = sysplex
            ===>                        ===>              minus systems in the
            ===>                        ===>              sysplex explicitly
            ===>                        ===>              defined in the SCDS
            ===>                        ===>
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 17. Defining Volume System Status

In the SMS Volume Status Define panel, you define the relationship between the DASD volume in the pool storage group and each system in the SMS complex. The volumes defined to the storage group are listed in Volume Serial Numbers and the systems you defined in the base configuration are listed under Systems.

SMS Volume Status lists the relationship between a DASD volume and each system in the SMS complex. DASD volumes have both a physical (actual) and a logical (system-defined) connection to systems in the SMS complex. You are responsible for maintaining consistency between these two types of connections.

For a system to have access to a data set, you need to define one of the five types of access from the system to a storage group, and you need to define one of the five types of access from the storage group to a DASD volume (ENABLE, DISALL, DISNEW, QUIALL, and QUINEW for both types of access). You must ensure that a physical connection exists from the DASD volume to the system, and the MVS status of the DASD volume must be ONLINE.

See “Defining System Access to Pool and VIO Storage Groups” on page 48 for an explanation of the SMS statuses. The volume statuses show the relationships between each system in the SMS complex and a given DASD volume in the pool storage group.

When a system attempts to allocate a data set, it proceeds in order through the following checks with respect to the current system:
1. Storage group status-ENABLE/DISALL/DISNEW/QUIALL/QUINEW/NOTCON
2. SMS volume status-ENABLE/DISALL/DISNEW/QUIALL/QUINEW/NOTCON
3. MVS volume status-ONLINE/OFFLINE

After defining the status of the DASD volume, issue the END command to return to the Storage Group Application Selection panel. You can define additional DASD volumes to the pool storage group, if you wish.

Assigning DASD Storage Groups to Data Sets

Storage groups can only be assigned through the storage group ACS routine. For a given data set, the storage group ACS routine runs only if the storage class ACS routine assigns a valid storage class. If the storage class is not valid, allocation fails. If no storage class is assigned, then the data set is not system-managed and is allocated according to the rules in a non-system-managed environment.

You can assign candidate storage groups to a data set allocation from which SMS selects eligible DASD volumes for the data sets. If the storage group ACS routine does not determine a storage group for a data set, allocation fails.

Restrictions: SMS does not check or verify DASDVOL authorization for data set allocations on SMS-managed volumes. By adding a volume to SMS, you are removing it from any active DASDVOL authorizations because DASDVOL access is not verified on SMS-managed volumes. See *z/OS Security Server RACF Security Administrator’s Guide* for further information on DASDVOL authority.

Defining an Object Storage Group

Before you define an object storage group that uses optical volumes, you should define your optical libraries. See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information.

Figure 18 on page 58 shows the Object Storage Group Define panel.

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

SCDS Name . . . . . : USER8.TEST.SCDS
Storage Group Name  : SGO

To DEFINE Storage Group, Specify:
  Description ==>
              ==>
  Qualifier  . . . . .           (1 to 8 character qualifier)
  Cycle Start Time . .       (0-23 or NONE)  End Time . .     (0-23 or blank)
  OSMC Processing System . . . .           (? for list of OSMC System names)

  Library Names   (1 to 8 Characters each):
   ===>           ===>           ===>           ===>
   ===>           ===>           ===>           ===>
  Volume Full Threshold  . . . .       (0-9999)
  Drive Start Threshold  . . . .       (0-9999)
  Volume Full at Write Error . .       (Y or N)
   DEFINE SMS Storage Group Status . .... Y  (DEFINE - Y, ALTER - 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 18. Defining Object Storage Group Attributes

The SCDS Name and Storage Group Name fields are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the object storage group.

To define an object storage group you must specify the following attributes:

Qualifier

This identifies a DB2 database on DASD which contains table spaces, one of which contains the object directory for this object storage hierarchy. The qualifier attribute also points to the DB2 object storage tables on DASD which have been defined for the object storage hierarchy. You can use one to eight alphanumeric characters as qualifiers.

This qualifier must be defined as a package in the CBRPBIND job to create the necessary DB2 package for the application plans.

Cycle Start Time

This is the beginning of a window of time when object processing can be started for this storage group. You must specify a value from 0 - 23, or NONE. A value from 0 - 23 represents an hour of the day. Specify an hour of the day as 00 for midnight; 01 for 1 a.m.; 23 for 11 p.m. for example. The hour of the day value for Cycle Start Time must be different from the value for Cycle End Time. Specify NONE if you do not want automatic processing for the storage group. When NONE is specified, the Cycle End Time value must be blank.

If you are running in an OAMplex and specify a cycle start time, we recommend that you also specify an OSMC processing system name, to avoid starting object processing for that object storage group on all systems in the OAMplex at cycle start time.

Cycle End Time

This is the end of a window of time when object processing can be started for this storage group. You must specify a value from 0 - 23, or blank (depending on what you specified for Cycle Start Time). A value of from 0 - 23 represents an hour of the day, and is required when a value from 0 to 23 was specified for Cycle Start Time. Specify an hour of the day as 00 for midnight; 01 for 1 a.m.; 23 for 11 p.m., for example. The hour of the day value for Cycle End Time must be different from the hour of the day value for Cycle Start Time. Leave the field blank if you specified NONE for Cycle Start Time.

OSMC Processing System

Specifies an OSMC processing system name, which identifies which OAM system in the sysplex performs OSMC processing for this object storage group. This prevents multiple systems from trying to process the same storage group at the same time, especially when the Cycle Start Time is used for automatically starting storage group processing.

Library Names

You can specify either one to eight real library names that represent optical libraries, or one to eight pseudo library names. The libraries you specify are used to process write requests to the storage group. Names must be valid real or pseudo optical library names defined in the SCDS.

Volume Full Threshold

This is the number of free KB triggering volume full processing for an optical volume within the object storage group. When the number of free KB falls below the threshold, the object access method marks the optical volume full. If the threshold is reached while OAM is writing an object, it continues writing that object until finished. OAM then writes no more objects to that volume. Valid values are from 0 to 9999.

Drive Start Threshold

This is the maximum number of object write requests outstanding for an optical drive in this storage group. When the number of object write requests to this storage group divided by the number of optical drives currently processing write requests for this storage group exceeds this threshold, the object access method attempts to start an additional optical drive. Valid values are from 0 to 9999.

This field is optional. If no value is specified, the default is 17.

Volume Full at Write Error This field indicates when to mark as “full” optical or tape volumes within this object storage group. If Y, the object access method marks an optical or tape volume full the first time an attempt to write an object on the volume fails because not enough space remains on the volume. If N, OAM marks a volume full only when the number of available KB in the user data area falls below the Volume Full Threshold for optical or the TAPEFULL THRESHOLD for tape. An optical volume is also marked full if the optical volume table of contents area is full, regardless of how many KB are remaining in the user data area.

SMS Storage Group Status This required field indicates whether you want to change the storage group status. If you are defining a storage group, the default value is Y. If you are altering a storage group, the default value is N and can be changed to Y or N.

You must define object storage group system statuses on the SMS Storage Group Status Define panel before the object storage group definition can be saved. After you define the status of the systems, issue the END command to save the defined object backup storage group and return to the Storage Group Application Selection panel.

Object tape parameters for object and object backup storage groups are specified via SETOAM statements in the CBROAM*xx* member of PARMLIB. For more information, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Defining the Object Backup Storage Groups

When you define object backup storage groups, you are naming the groups that are used to contain backup data and optionally specifying the names of optical libraries that can contain backup copies of objects. There can be one or two object backup storage groups per object storage group. Figure 19 on page 61 shows the Object Backup Storage Group Define panel.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                       OBJECT BACKUP STORAGE GROUP DEFINE
Command ===>

SCDS Name . . . . . : SMS.SCDS1.SCDS
Storage Group Name  : OBJB1

To DEFINE Storage Group, Specify:

  Description ==>
              ==>
  Library Names   (1 to 8 Characters each):
   ===>           ===>           ===>           ===>
   ===>           ===>           ===>           ===>

  Volume Full Threshold  . . . .       (0-9999)
  Drive Start Threshold  . . . .       (0-9999)
  Volume Full at Write Error . .       (Y or N)

   SMS Storage Group Status . . . . . . . Y  (DEFINE - Y, ALTER - 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 19. Defining Object Backup Storage Group Attributes

The SCDS Name and Storage Group Name fields are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the object backup storage group.

To define an object backup storage group you must specify the following attributes:

Library Names
You can specify up to eight real library names that represent optical libraries, or one to eight pseudo library names. The libraries you specify are used to process write requests to the storage group. Names must be valid real or pseudo optical library names defined in the SCDS.

Volume Full Threshold
The number of free KB triggering volume full processing for an optical volume within the object backup storage group. When the number of free KB falls below the threshold, the object access method marks the optical volume full. If the threshold is reached while OAM is writing an object, OAM continues writing that object until finished. OAM then writes no more objects to that volume. Valid values are from 0 to 9999. This is a required field if a library name is specified.

Drive Start Threshold
This is the maximum number of object write requests outstanding for an optical drive in this storage group. When the number of object write requests to this storage group divided by the number of optical drives currently processing requests for this storage group exceeds this threshold, the object access method attempts to start an additional optical drive. Valid values are from 0 to 9999. This is a required field if a library name is specified.

Volume Full at Write Error
This field indicates when to mark as “full” optical volumes within this backup object storage group. If Y, the OAM marks an optical volume full the first time an attempt to write an object on the optical volume fails because not enough space remains on the optical volume. If N, the object access method marks an optical full only when the number of available KB in the user data area falls below the Volume Full Threshold. The volume is also marked full if the optical volume table of contents area is full, regardless of how many KB are remaining in the user data area. This is a required field if a library name is specified.

SMS Storage Group Status
This required field indicates whether you want to change the storage group status. If you are defining a storage group, the default value is Y and cannot be changed to N. If you are altering a storage group, the default value is N and can be changed to Y or N.

You must define object backup storage group system statuses on the SMS Storage Group Status Define panel before the object backup storage group definition can be saved. After you define the system statuses, issue the END command to save the newly defined object storage group and return to the Storage Group Application Selection panel.

Object tape parameters for object and object backup storage groups are specified via SETOAM statements in the CBROAM*xx* member of PARMLIB. For more information, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Defining Object or Object Backup Storage Group Status

After you have supplied all required fields on the Object Storage Group Define panel or the Object Backup Storage Group Define panel with valid values, press Enter. The SMS Storage Group Status Define panel shown in Figure 20 on page 63 is displayed.

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

SCDS Name  . . . . : SMS.SCDS1.SCDS
Storage Group Name : OBJ1
Storage Group Type : OBJECT

To DEFINE Storage Group System Status, Specify:

  System           SMS SG       System         SMS SG
  Name             Status       Name           Status
 --------          ------      --------        ------
 SYSTEM1      ===> ENABLE      SYSTEM2    ===> NOTCON   ( Possible SMS SG
 SYSTEM3      ===> NOTCON      SYSTEM4    ===> NOTCON     status value for
 SYSTEM5      ===> NOTCON      SYSTEM6    ===> NOTCON     each System: NOTCON,
              ===>                        ===>            ENABLE, DISALL,
              ===>                        ===>            or DISNEW )
              ===>                        ===>
              ===>                        ===>          ( Only system names
              ===>                        ===>            are displayed )
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 20. Defining Object or Object Backup Storage Group System Status

Only systems and not system groups are shown on this panel, because you cannot connect an object or object backup storage group to a system group.

The SCDS Name, Storage Group Name and Storage Group Type fields are output fields that contain the information you specified on the Storage Group Application Selection panel. The System Name field is also an output field. It lists the systems that you defined in the base configuration. The SMS Storage Group Status field lists the relationship between the object or object backup storage group and each system in the SMS complex.

The devices associated with an object or object backup storage group have both physical (actual) and logical (system defined) connections to systems in the SMS complex. You are responsible for maintaining consistency between these two types of connections.

One of the following four relationships exists between each system in the SMS complex and the object or object backup storage group:

ENABLE For object storage groups, SMS permits application access to the object storage hierarchy of this group. For object backup storage groups, SMS permits access to the volume set. All OSREQ functions are allowed.

DISALL For object or object backup storage groups, SMS permits restricted access to the object storage hierarchy. The OSREQ functions STORE, RETRIEVE, and DELETE are denied to applications. All object processing continues to be done for the storage group.

DISNEW For object or object backup storage groups, SMS permits restricted access to the hierarchy. The OSREQ function STORE is denied to applications. All object processing continues to be done for the storage group.

NOTCON The named system cannot process this object/object backup group. NOTCON is the default.

You can enable object and object backup storage groups to more than one system. If you are not in an OAMplex, OAM ignores object storage groups that are defined as being connected to more than one system and issues a message.

After establishing the relationships between the object or object backup storage group and the systems in the SMS complex, you need to complete the definition of the storage group. From the SMS Storage Group Status Define panel, enter the END command to save your values and return to the Object Storage Group or Object Backup Storage Group Define panel. On this panel, enter the END command to save the storage group.

Assigning an OAM Object Collection to a Storage Group

OAM object collections are always managed by SMS. The ACS routines are executed when the OAM object collection catalog entry is initially defined. The ACS routines provide storage class and management class names that are used as the default storage class and management class assignments recorded in the catalog entry, and the storage group assignment for the collection. OAM uses the storage group assignment to identify the DB2 tables used for storage of the object directory information, for storage of objects on DASD, and to select the optical or tape volume used to store objects that belong to the collection.

Defining a Tape Storage Group

Figure 21 shows the Tape Storage Group Define panel. The SCDS Name and Storage Group Name fields are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the tape storage group. ISMF primes the field with blanks, which are the default. Issue the END command to save the newly defined tape storage group and return to the Storage Group Application Selection panel.

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

SCDS Name . . . . . : SMS.SCDS1.SCDS
Storage Group Name  : TAPE1

To DEFINE Storage Group, Specify:

  Description ==>
              ==>

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



  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 21. Defining Tape Storage Group Attributes

The Library Names field is used to specify the tape libraries that own the volumes within this storage group. One to eight library names can be associated with a tape storage group. At least one library name must be specified when defining a tape storage group. The library name in the tape storage group definition must also be defined in the same SCDS.

The SMS Storage Group Status field is used to specify whether or not you want to modify the status (enable, disall, disnew, notcon) of this storage group for each one of the systems in the SMS complex. If Y is specified, the SMS Storage Group Status Define panel is displayed. If N is specified, the status panel is not displayed, and the storage group defaults to ENABLE for each one of the systems in the complex. This is a required field. N is the default.

Defining Tape Storage Group Status

In the Define SMS Storage Group Status field, indicate whether you want to change the status of the tape storage group with respect to a given system in the SMS complex. Initially, all of the status fields are ENABLE. Specify Y, yes, to change a status. Accept the default N, no, to leave all the status fields as ENABLE.

When you have completed all your entries in the Tape Storage Group Define panel, press Enter. If you specified Y for the Define SMS Storage Group Status attribute, you get the SMS Storage Group Status Define panel shown in Figure 22, where you can define the status of the storage group to each system in the SMS complex.

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

SCDS Name  . . . . : SMS.SCDS1.SCDS
Storage Group Name : TAPE1
Storage Group Type : TAPE
To DEFINE Storage Group System/                            ( Possible SMS SG
Sys Group Status, Specify:                               Status for each:
                                                         - Pool SG Type
 System/Sys      SMS SG    System/Sys      SMS SG          NOTCON, ENABLE
 Group Name      Status    Group Name      Status          DISALL, DISNEW
 ----------      ------    ----------      ------          QUIALL, QUINEW
 *SYSPLX1   ===> ENABLE    SYSTEM2    ===> ENABLE        - Tape SG Type
 SYSTEM3    ===> NOTCON    SYSTEM4    ===> NOTCON          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 22. Defining Tape Storage Group System Status

The SCDS Name, Storage Group Name and Storage Group Type fields are output fields that contain the information you specified on the Storage Group Application Selection panel. The System/Sys Group Name field is also an output field. It lists the systems that you defined in the base configuration (using the CDS application). The SMS Storage Group Status field lists the relationship between the storage group and each system in the SMS complex. See “Defining System Access to Pool and VIO Storage Groups” on page 48 for an explanation of these relationships.

From the SMS Storage Group Status Define panel, enter the END command to save your values and return to the Tape Storage Group Define panel. On this panel, enter the END command to save the tape storage group. This returns you to the Storage Group Application Selection panel, which is primed with your CDS and storage group name.

Defining Additional Storage Groups

You can copy existing storage groups and modify them to create new storage groups by using the COPY line operator, which is explained in “Copying SMS Classes, Storage Groups, and Aggregate Groups” on page 209.

When you copy either a pool or VIO storage group within an SCDS, all of the system status values and system names for automatic processing are copied. When you copy either a pool or VIO storage group from one SCDS to a different SCDS, all of the system status values default to ENABLE rather than to the system status values of the source storage group. Also, the Migrate System Name, Backup System Name, and Dump System Name fields are set to blank.

Whenever you copy an object storage group within an SCDS, you must modify the qualifier to ensure that the new object storage group has a unique qualifier for the connected system. All qualifiers within an system must be unique. Specify Y for the Perform Alter attribute on the Copy Entry panel to display the Object Storage Group Alter panel. Modify the qualifier and issue an END command to save the new qualifier.

Whenever you copy an object storage group from one SCDS into a different SCDS, the qualifier must be unique within the target SCDS. If necessary, alter the qualifier in the same manner as required for copying an object storage group within an SCDS.

The system status values of object and object backup storage groups are retained when copying within an SCDS. Object and object backup storage groups can be enabled by and connected to more than one system. Specify Y for the Perform Alter attribute on the Copy Entry panel, and Y for the Alter SMS Storage Group Status attribute on either the Object Storage Group Alter panel or Object Backup Storage Group Alter panel to display the system statuses for modification.

Listing Volumes in a Storage Group

Select option 1, List, on the Storage Group Application Selection panel, Figure 11 on page 45, to get a listing of the volumes in the storage group specified in the panel. Enter the LISTVOL line operator next to the storage group to select a list. The LISTVOL line operator allows you to list volume information, such as volume serial numbers, for a pool, dummy, object, object backup storage group or tape storage group. You can use the LISTVOL storage group line operator without any parameters to list all volumes in a specified storage group. See the example shown in Table 2 on page 67. The list displayed depends on the storage group type.

The DASD Volume STATUS line operator allows users to display 32 sets of SMS and MVS volume statuses corresponding to 32 system names, system group names or both for a volume from the Volume List panel. The Volume List panel only displays eight of the possible 32 sets of SMS and MVS volume statuses at a time. In order to view volume statuses beyond the first eight sets, this line operator has to be entered against the volume on the Volume List Panel. This line operator STATUS can be abbreviated to a minimum of the first two characters ″ST″.

Table 2. Volumes Listed in a Specified Storage Group

Storage Group Type	Volume List Shown
POOL DUMMY	DASD Volume List
OBJECT or OBJECT BACKUP	Mountable Optical Volume List
TAPE	Mountable Tape Volume List

Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for information on the Mountable Optical Volume List and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for information on the Mountable Tape Volume List.

Chapter 5. Defining Management Classes

DFSMShsm manages non-SMS data sets at the volume level, applying the management criteria to all data sets on a given volume. SMS automates the management of storage at the data set level by introducing management classes. When assigned to data sets, management classes replace and expand attributes that otherwise would be specified on JCL DD statements, IDCAMS DEFINE commands, and DFSMShsm commands. When assigned to objects, OAM uses a subset of the management class attributes and the OSMC class transition attributes to manage objects. This chapter describes management classes and shows you how to define them using the ISMF management class application.

With the Copy Technique attribute, the storage administrator has the option of specifying whether a system-managed backup process called concurrent copy should be used for data sets to enhance system availability during data set backup and aggregate backup processing. Data Set Alter can be used to alter a management class so data sets can use the concurrent copy technique during backup processing. The storage administrator has the option of specifying whether the concurrent copy process is required, preferred, or discouraged. The Copy Technique attribute is related to the Backup and Aggregate Backup components of management class. A Copy Technique field exists for each of the Backup and Aggregate Backup components. These fields appear on the management class list, display, define/alter, view, and sort panels.

Understanding Management Classes

A management class is a list of data set migration, backup and retention attribute values. Management class also includes object expiration criteria, object backup requirements, and class transition criteria for management of objects. DFSMShsm uses the attributes of the management class associated with a data set to manage storage. When you assign a management class to a system-managed DASD data set, SMS places the management class name in both the basic catalog structure (BCS) and the VSAM volume data sets (VVDS) catalog entries of the data set. Management class is optional for system-managed data sets and does not apply to non-system-managed data sets. Management class is *not* used for tape data sets.

If you alter a management class definition, SMS applies the changes to any new data sets or objects after you activate the changed configuration. SMS also applies the changes to any previously allocated data sets or objects, beginning with their next scheduled management cycle (such as daily space management or backup).

Default Management Class

For data sets, you can specify a default management class in an SCDS. DFSMShsm applies the default management class to all system-managed data sets that do not already have a management class. Unlike the data sets that you have already assigned a management class, the catalog entries of these data sets do not contain a management class name. For objects, the default management class is defined in the catalog entry for the object collection to which the objects belong. The default management class is assigned by the management class ACS routine when the first object is stored in the collection.

© Copyright IBM Corp. 1986, 2002

OAM Management Classes

OAM uses some attributes in the management class associated with the object to manage the object. Class transition attributes allow OAM to change the way an object is managed based on its age, its usage, or a predefined periodic function. A class transition is a change in the object's management class or storage class when an event occurs which brings about a change in an object's management criteria or service level. Class transitions occur during an OSMC storage management cycle. Objects requiring class transition use the ACS routines to determine if they should be managed using a different management class or placed at a different level of the storage hierarchy according to a new storage class.

OAM uses the backup attributes of the management class definition to initiate the writing of one or two backup copies of an object. The copy or copies are made during the first storage management cycle after the object has been stored and a management class with backup attributes has been assigned to the object. If two backup copies are desired, specify the SETOSMC statements in your CBROAM*xx* member of PARMLIB using SECONDBACKUPGROUP to invoke multiple object backup support.

Tip: The default number of backup copies is two. If you want only one backup copy, you must set the management class definition for the number of backup versions to one.

Retention attributes determine the OAM action for object expiration. An object can expire automatically based on its age, its usage, or a specific date derived from its management class and an object-specific retention period, if provided. OSMC deletes expired objects from the directory automatically during the storage management cycle with the approval of the auto-delete installation exit.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for a more specific discussion of how to use management classes with objects.

Describing Management Classes

A management class definition contains both descriptive and storage-related information. To identify and refer to your management classes, you assign each one a unique name that contains from one to eight alphanumeric characters. Each management class definition maintains an owner ID that identifies the storage administrator who originally created or last modified the management class. The *owner ID* is a z/OS user ID on the ISMF Management Class List in column 19-Last Modified Userid. Also, each management class contains an optional 120 character description field for describing its contents.

Planning Management Classes

You should create a management class for each type of service that is to be provided by the installation. A type of service is defined for a collection of data sets that have similar migration, backup, and retention requirements or objects that have similar backup, expiration, and class transition requirements. Before you actually define your management classes, you should gather the following information:

- Requirements for releasing over-allocated space
- Migration requirements
- Retention criteria
- Treatment of expired data sets
- Frequency of backup
- Number of backup versions
- Retention of backup versions
- Number versions
- Retain only version
- Retain only version unit
- Retain extra versions
- Retain extra versions unit
- Copy serialization
- Generation data group (GDG) information
- Object class transition criteria

Based on this information, you can establish management classes that centralize storage management in the SMS complex.

If a data set has a management class that specifies automatic backup or migration, then you must direct the data set to storage groups that are eligible to be processed for automatic backup or migration capabilities.

A generation data group (GDG) is a group of related cataloged data sets that have sequentially ordered names. SMS uses GDG-related information in the management class definition to manage the storage associated with these data sets. Some management criteria is specified in the definition of the GDG. Some management criteria might be specified by assigning a management class to each individual generation in the GDG. Generation data sets within the same GDG might have different management classes assigned by JCL.

DFSMSrmm uses management class names, instead of attributes, for tape data set retention and movement. It matches a management class name with a DFSMSrmm vital record specification (VRS) and implements the policies defined in the VRS.

Defining Management Class Attributes

You can use ISMF to define your management classes by selecting option 3, Management Class, from the ISMF Primary Option Menu for storage administrators. Figure 23 on page 72 and Figure 24 on page 73 illustrate the Management Class Application Selection panels.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                     MANAGEMENT CLASS APPLICATION SELECTION         Page 1 of 2
Command ===>

To perform Management Class Operations, Specify:
  CDS Name  . . . . . . . . . 'SMS.SCDS1.SCDS'
                               (1 to 44 character data set name or 'Active' )
  Management Class Name . . . INTERIM   (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 23. Management Class Application Selection Panel for Storage Administrators, Page 1 of 2

See the following topics or publications for further information on the options:

- "Listing SMS Classes, Aggregate Groups, Storage Groups, and Libraries Using ISMF" on page 195 for more information on option 1
- *z/OS DFSMS: Using the Interactive Storage Management Facility* for more information on option 2
- "Altering Management Classes" on page 202 for more information on option 4

To define a management class, you must specify a CDS Name and a Management Class Name on the panel and select option 3, Define. The CDS Name must be the name of an SCDS. ISMF primes the field with the last used name. The default is the quoted word 'ACTIVE', which represents the currently active configuration, but you cannot define or alter management classes to the 'ACTIVE' configuration.

In the Management Class Name field, you must specify the name of the management class that you are defining. Your management class names should be generic rather than specific, so that attribute changes do not make names meaningless or misleading. STANDARD and INTERIM are examples of generic names. RETAIN30 and BUDAILY are examples of specific names. ISMF primes the field with the name last used.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                     MANAGEMENT CLASS APPLICATION SELECTION         Page 2 of 2
Command ===>

CDS Name  . . . . . . . : SMS.SCDS1.SCDS
Management Class Name . : INTERIM


If option other than List is chosen in in the previous page, Select attribute
group to Display, Define or Alter first :

   1  1. Space Management Attributes  -  Pages 1 and 2
      2. Backup Attributes            -  Page  3
      3. Class Transition Attributes  -  Page  4
      4. Aggregate Backup Attributes  -  Page  5





Use ENTER to Perform Selection; Use UP Command for previous Selection Panel;
Use HELP Command for Help; Use END Command to Exit;
```

Figure 24. Management Class Application Selection Panel for Storage Administrators, Page 2 of 2

Designation of an attribute group is only useful when performing a display, define, or alter function. An attribute group is selected for processing ahead of other Management Class attributes by entering its corresponding number in the selection field provided. Panels containing the other management class attributes can be accessed by paging up and down in the usual manner.

Defining Management Class Expiration Attributes

Figure 25 on page 74 shows the first page of the Management Class Define panel. You can leave any page of the Management Class Define panel without saving the management class by issuing the Cancel command.

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

SCDS Name . . . . . . : SMS.SCDS1.SCDS
Management Class Name : INTERIM

To DEFINE Management Class, Specify:

  Description ==>
              ==>

  Expiration Attributes
    Expire after Days Non-usage  . . 2           (1 to 9999 or NOLIMIT)
    Expire after Date/Days . . . . . 7           (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 25. Defining Management Class Expiration Attributes

Page one contains the management class expiration attributes. DFSMShsm processes the expiration attributes before the migration attributes that you specify on the second page of the Management Class Define panel. SCDS Name and Management Class Name are output fields that contain the SCDS and management class names you specified in the Management Class Application Selection panel. Description is an optional field of 120 characters where you can describe the management class.

Expiration Attributes

You use expiration attributes to determine the action for data set and object expiration and deletion. DFSMShsm deletes expired data sets during automatic space management processing. OAM deletes objects during a storage management cycle with the approval of the auto-delete installation exit. Expiration attributes are required values that indicate when a data set or object becomes eligible for expiration. You can base expiration criteria on a specific date, on the number of days since the data set or object was allocated, or on the number of days since the data set or object was last referenced.

The Expire after Days Non-usage field specifies how much time must elapse since last access before a data set or object becomes eligible for expiration. The Expire after Date/Days field specifies an absolute date or period after its allocation for a data set or object to become eligible for expiration. The default for both fields is NOLIMIT.

Retention Limit

The Retention Limit value is a required value that limits the use of retention period (RETPD) and expiration date (EXPDT) values that are explicitly specified in JCL, are derived from management class definitions or are explicitly specified in the OSREQ STORE macro. If the value of a user-specified RETPD or EXPDT is within the limits specified in the Retention Limit field, it is saved for the data set. For objects, only RETPD is saved.

The default retention limit is NOLIMIT. If you specify zero, a user-specified or data class derived EXPDT or RETPD is ignored. If users specify values that exceed the maximum period, the retention limit value overrides not only their values but also the expiration attributes values. The retention limit value is saved. ISMF primes the Retention Limit field with what you specified the last time.

The Use of Expiration Attributes and Retention Limit

DFSMShsm determines if a data set has expired based on the expiration date found in the catalog entry of the data set. OAM determines if an object has expired based on the expiration date in its object directory entry. If an expiration date is not found, DFSMShsm and OAM use the management class expiration attributes. These attributes are used as follows:

- If both expiration attributes are NOLIMIT, the data set or object never expires.
- If one of the expiration attributes is NOLIMIT, then the other attribute must be satisfied.
- If neither expiration attribute is NOLIMIT, both of the expiration attributes must be satisfied.

Tip: If you want to change the expiration date in a catalog entry, you can either use the access method services ALTER command or you can specify either the RETPD or EXPDT parameter in the JCL for an old managed non-VSAM data set, as long as the new expiration date is valid based on the retention limit specified. See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for information on changing the expiration date of an object.

Data sets or objects having the INTERIM management class defined in Figure 25 on page 74 become eligible for expiration when both of the following criteria are met: at least seven days since allocation and not referenced in the last two days.

Table 3 shows several combinations of retention attributes used for space management processing. Use the highlighted values for each instance.

Table 3. Comparing Retention Period Attributes

Attribute	Case 1	Case 2	Case 3	Case 4	Case 5
Retention Limit	0	**50**	100	NOLIMIT	100
Expire after Days Non-usage	**50**	50	50	50	50
Expire after Date/Days	**100**	100	100	100	**100**
RETPD/EXPDT	60	60	**60**	**60**	

In the first case, the retention limit is zero, so DFSMShsm and OAM honor the values of 50 and 100 which are the values specified for Expire after Days Non-usage and Expire after Date/Days and ignore any user-specified or data class values. In Case 2, the management class expiration values are ignored because RETPD and EXPDT values have been specified or derived and the retention limit is nonzero. However, because the retention limit is less than the user-specified or data class values, 50 is saved and used to calculate the expiration date.

In Case 3, the user-specified or data class values fall within the retention limit. So, DFSMShsm uses the values of 60 and 60. OAM uses a RETPD value of 60, because OAM does not consider EXPDT.

In Case 4, the RETPD and EXPDT are used because the retention period value is NOLIMIT.

In Case 5, because no user-specified or data class derived values are available, DFSMShsm and OAM use the values specified in the management class expiration attributes which are shown on the Management Class Define panel.

After you specify the expiration attributes, issue the DOWN command to see the next page of the Management Class Define panel, which is shown in Figure 26, on which you can specify the migration attributes.

Defining Management Class Migration Attributes

To DFSMShsm, a data set occupies one of two distinct states in storage:

Primary Also known as *level 0*, the primary state indicates that end users can directly access a data set residing on a volume.

Migrated End users cannot directly access data sets that have migrated from the primary state to a migrated state. To be accessed, the data sets must be recalled to primary storage. A migrated data set can reside on either *migration level 1* (usually permanently mounted DASD) or *migration level 2* (usually tape).

A data set can move back and forth between these two states, and it can move from level 0 to migration level 2 (and back) without passing through migration level 1. Objects do not migrate. Movement back to level 0 is called *recall*.

Figure 26 shows the second page of the Management Class Define panel, which contains the management class migration and GDG management attributes. SCDS Name and Management Class Name are output fields that contain the SCDS and management class names you specified in the Management Class Application Selection panel.

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

SCDS Name . . . . . . : SMS.SCD1.SCDS
Management Class Name : INTERIM

To DEFINE Management Class, Specify:

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

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


  GDG Management Attributes
    # GDG Elements on Primary . . . 1           (0 to 255 or blank)
    Rolled-off GDS Action . . . . . EXPIRE      (MIGRATE, EXPIRE or blank)

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

Figure 26. Defining Management Class Migration Attributes

To define a management class, you must specify values for the Partial Release and Command or Auto Migrate attributes. If you specify BOTH for the Command or Auto Migrate attribute, then Primary Days Non-usage and Level 1 Days Non-usage become required fields. If you specify NONE for the Command or Auto Migrate attribute, the expiration attributes specified on page 1 of this panel still apply. The other fields on this panel are optional.

Partial Release Attribute

The Partial Release attribute applies to non-VSAM data sets and to all VSAM data set types allocated in the extended format. You can use partial release to choose the conditions under which unused allocated space is released. If you select partial release, space release is carried out automatically at the time you have selected. Each partial release option releases the same amount of space. Unused space is released in cylinders or tracks, depending on the space allocation unit used. The following options are available for partial release:

Y Yes. Release unused space at Space Management cycle time.

YI Yes Immediate. Release unused space at Space Management cycle time and also when close is issued for a data set that was open for output.

C Conditional. If a nonzero secondary space allocation has been specified for the data set, release unused space at Space Management cycle time. This option only applies to physical sequential and partitioned data sets. If specified for an extended format VSAM data set, this option is processed as if Yes were specified.

CI Conditional Immediate. If secondary space has been allocated, release unused space at Space Management cycle time and also when close is issued for a data set that was open for output.

N No. No release of unused space.

Notes:

1. ISMF primes the field with the last used value. The default is N.
2. The Partial Release attribute is not checked during DFSMShsm interval migration.
3. The Partial Release attribute is mutually exclusive with the Guaranteed Space storage class attribute.
4. Partial release is ignored under the following conditions:
   - Another job is sharing the data set (DISP=SHR).
   - Another task in the same job is processing an OPEN, CLOSE, EOV, or FEOV request for the data set.
   - Another DCB is open for the data set.

Related Reading: For more information on partial release, see *z/OS DFSMS: Using Data Sets*.

Migration Attributes

The Primary Days Non-Usage attribute represents the minimum number of days that must elapse since last access before a data set is eligible for normal migration. Days refers to calendar days, not to a 24-hour time period. In other words, the day you close the data set counts as the first full day of nonusage. The minimum number of Primary Days Non-usage days required for migration to ML2 includes the time already spent unreferenced on ML0. The default is 2. (See “GDG Management Attributes” on page 78 for the special GDS migration eligibility rules.)

A nonzero number for Level 1 Days Non-usage indicates the number of days that must elapse since the last reference on level 0 before a data set on ML1 can migrate from level 1 to level 2. The default is 60.

The Command of Auto Migrate attribute allows you to specify if a data set is eligible to be migrated by BOTH command and automatic processing, by command alone, or not at all (NONE).

If you do not want to migrate data sets that belong to a particular management class, specify NONE in the Command or Auto Migrate field. The data sets remain on primary storage until they expire. If you want data sets to be eligible for migration directly from level 0 storage to level 2 tape, specify 0 in the Level 1 Days Non-usage field. Otherwise, the data sets must first migrate to level 1 (the days spent on level 0 count toward the eligibility of moving to level 2).

GDG Management Attributes

The GDG Management Attributes indicate criteria for early migration of GDGs off primary storage and what to do with rolled-off generation data sets (GDSs).

The # GDG Elements on Primary attribute indicates how many of the most recent generations of a GDG use the normal Primary Days Non-usage attribute for migration criteria. Generations that are over this limit are eligible for early migration during the next processing of primary space management. The inactive age is irrelevant for these 'over the limit' GDSs and they are given a higher selection priority so they are likely to be migrated if space is needed to meet volume thresholds.

Clarification: Specifying the # GDG Elements on Primary attribute does not ensure that the specified number of GDSs are kept on primary storage, but rather that their migration is not accelerated.

To specify that old generations are to have priority for migration and to specify how many generations are to have standard priority, you can specify the # GDG Elements on Primary attribute. When you specify this attribute, all non-rolled-off generations of each GDG that exceed the number you specify are given priority for early migration. If you specify 0 for this attribute, all generations of the GDGs are given priority for early migration. Those data sets made eligible for early migration do not have to satisfy the Primary Days Non-usage criteria.

All generations that meet the Primary Days Non-usage criteria are eligible to migrate. # GDG Elements on Primary enables you to accelerate migration of older generations while Primary Days Non-usage enables you to specify migration criteria of younger generations separately.

The Rolled-off GDS Action value indicates whether to expire rolled-off GDSs or to make them eligible for migration. If you specify MIGRATE, management class expiration attributes are applied to the data set to determine if the data set should be deleted; if not, the data set is eligible for migration. If you specify EXPIRE, the rolled-off GDS is deleted unless an explicit expiration date is in the DSCB. In this case, the rolled off GDS becomes a non-GDG data set and migrates according to non-GDG migration attributes.

Tip: The DFSMShsm parameter, EXPIREDATASET(SCR | NOSCR), controls the deletion of a data set with an expiration date in the DSCB.

Both the # GDG Elements on Primary and the Rolled-off GDS Action attributes are optional and have default values of blank. If left blank, no special treatment results from the data set being a GDS or a rolled-off GDS. That is, other management class attributes is used to process the data set.

During automatic primary space management, GDSs that have been determined as eligible for migration are migrated in the following priority order:

1. GDSs that are rolled-off.
2. GDSs that exceed the limit specified in # GDG Elements on Primary.
3. Other GDSs are handled by the non-GDG algorithm which bases priority on a function of size and the length of time it has been eligible to migrate in Primary Days Non-usage.

After specifying the migration attributes, issue the DOWN command to specify the backup attributes on the next page of the Management Class Define panel, which is shown in Figure 27.

Defining Management Class Backup Attributes

Figure 27 shows the third page of the Management Class Define panel, which contains the management class backup attributes. (For objects, see “Defining Object Class Transistion Attributes” on page 81.)

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

SCDS Name . . . . . . : SMS.SCDS1.SCDS
Management Class Name : INTERIM

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  . . NONE     (BOTH, ADMIN or NONE)
    Auto Backup . . . . . . . . . . . Y        (Y or N)
    Backup Copy Technique . . . . . . S        (P=Conc Preferred, R=Conc
                                               Required or S=Standard)
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 27. Defining Management Class Backup Attributes

Backup Frequency

The Backup Frequency attribute specifies how many days must elapse before DFSMShsm can back up data sets that have changed since the last backup. If you want to back up changed data sets every time that DFSMShsm processes the volumes containing them, specify 0. The default is 1.

This field does not apply to objects.

Number of Backup Versions

The Number of Backup Versions fields specify the maximum number of backup versions to retain for a data set. The default is 2 if the data set still exists and 1 if it has been deleted.

The number of backup versions is used to determine whether OAM should write one or two backup copies of the objects, when you activate the SECONDBACKUPGROUP function for objects using SETOSMC in the CBROAM*xx* member of PARMLIB. If the number of backup versions is greater than 1 and AUTO BACKUP is Y, OAM will create two backup copies. When the original object is expired or deleted, all backup copies are also deleted.

Retain Days Only Backup Versions

The Retain Days Only Backup Version (Data Set Deleted) field indicates how many days to keep the most recent backup version of a deleted data set, starting from the day DFSMShsm detects it has been deleted. This attribute applies only when a data set no longer exists on primary (level 0) or migrated (levels 1 and 2) storage. The default is 60.

This field does not apply to objects. Backup copies of objects are not retained when the original object is deleted.

Retain Days Extra Backup Versions

The Retain Days Extra Backup versions field indicates how many days to keep backup versions other than the most recent one, starting from the day backups were created. It only applies when more than one backup version exists, and when a data set has low activity. When a new backup version is created and the number of backup versions already equals the value specified in the pertinent (existing or deleted) Number of Backup Versions field, the oldest version is deleted. This attribute applies whether the data set has been deleted or not. The default is 30.

Admin or User Command Backup

The Admin or User command Backup field indicates if both the end user and the storage administrator can issue command backups of the data sets in this management class, if only the storage administrator can, or if neither of them can. The Admin or User command Backup field is required, and has a default value of BOTH. Otherwise, the remaining fields are optional.

BOTH indicates that both storage administrators and users can perform command backups against these data sets.

ADMIN indicates that only storage administrators can perform command backups against these data sets.

NONE indicates that neither storage administrators or end users can perform command backups.

If you specify BOTH or ADMIN in the Admin or User Command Backup field, the remaining fields on this panel are required. If the value is BOTH or ADMIN, then AUTO BACKUP specifies whether these data sets are eligible for automatic backup.

Auto Backup

The Auto Backup field is required, and has a default value of Y. If you specify Y in the Auto Backup field, the remaining fields on this panel are required. Otherwise, the remaining fields are optional.

OAM backs up the object during the first management cycle after the object is assigned to a management class requiring backup. This can occur when the object is stored or when its management class is changed by you or by a class transition. Backup copies of an object are deleted when the object is deleted. No archived copy is saved.

Backup Copy Technique

The Backup Copy Technique field specifies whether concurrent copy should be used during data set backup processing.

When deciding whether or not to use concurrent copy for a particular dump or copy operation, you need to decide why you are making a copy of the data. If the intention of the copy is to capture a ″point-in-time″ image of the data at a specific time, concurrent copy might not be appropriate. When you use concurrent copy there is always a chance that a dump or copy operation does not complete once you have started to make updates to the data. If this happens, the specific ″point-in-time″ is lost and you might not be able to recover an image of the data at that time.

R indicates that concurrent copy must be used for backup. Backup fails for data sets that don't reside on volumes supported by concurrent copy, or that are unavailable for concurrent copy.

P indicates that concurrent copy is preferred and should be used for backup. A data set is backed up on a nonconcurrent copy volume if it does not reside on a volume supported by concurrent copy, or if the volume on which it resides is unavailable for concurrent copy.

S indicates standard allocation, in which data sets are backed up without using concurrent copy.

Guideline: Take special care to avoid specifying conflicting conditions in the storage class and the management class. For example, if you specify Backup Copy Technique as R (required) in the management class and Accessibility as NOPREF in the storage class, you would be requesting concurrent copy for backup while having placed the data set on a device that does not support concurrent copy, which would result in the backup failing. Specifying the backup copy technique option in the management class in no way affects the placement of data sets on a concurrent copy volume.

After you specify the backup attributes, issue the DOWN command to see the next page of the Management Class Define panel, which is shown in Figure 28 on page 82, on which you can specify the object attributes.

See *International Technical Support Centers Implementing Concurrent Copy* for more information.

Defining Object Class Transistion Attributes

Page 4 of the Management Class Define panel, shown in Figure 28 on page 82, contains the object class transition attributes.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                          MANAGEMENT CLASS DEFINE              Page 4 of 5
Command ===>

SCDS Name . . . . . . : SMS.SCDS1.SCDS
Management Class Name : INTERIM

To DEFINE Management Class, Specify:
  Object Class Transition Criteria
    Time Since Creation Years  . .       (0 to 9999 or blank)
                        Months . .       (0 to 9999 or blank)
                        Days . . .       (0 to 9999 or blank)
    Time Since Last Use Years  . .       (0 to 9999 or blank)
                        Months . .       (0 to 9999 or blank)
                        Days . . .       (0 to 9999 or blank)
    Periodic
      Monthly   On Day . . .        (1 to 31, FIRST, LAST or blank)
      Quarterly On Day . . .        (1 to 92, FIRST, LAST or blank)
                In Month . .        (1 to 3 or blank)
      Yearly    On Day . . .        (1 to 366, FIRST, LAST or blank)
                In Month . .        (1 to 12 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 28. Defining Object Class Transition Attributes

The Time Since Creation Years, Months, or Days fields indicate the time since the creation date that must pass before transition occurs.

The Time Since Last Use Years, Months, or Days fields indicate the time since the last reference date that must pass before transition occurs.

The Periodic field indicates a time based on the calendar at which transition occurs.

Restrictions: The Time Since Creation, Time Since Last Use and Periodic fields cannot be specified together. A maximum date of 9999/12/31 is used if the requested Time Since Creation or Time Since Last Used exceeds the maximum date.

The Monthly On Day field specifies the day of each month that the transition occurs. If there are fewer days in the month than the number specified, the transition occurs on the last day of the month.

The Quarterly On Day or In Month fields specify the time of each quarter that the transition occurs. If both Day and Month are specified, this attribute specifies the day of the month in each quarter that the transition occurs. If there are fewer days in the specified month than the number specified in Day, then the transition occurs on the last day of the specified month.

The Yearly On Day or In Month fields specify the day or month of each year that transition occurs. If both Day and Month are specified, this attribute specifies the day of the month in each year that transition occurs.

FIRST specifies that the transition occurs on the first day of each month, quarter, or year, whichever attribute is specified.

LAST specifies that the transition occurs on the last day of each month, quarter, or year.

Defining Aggregate Backup Attributes

The fifth page of the Management Class Define panel shown in Figure 29 contains the aggregate backup attributes.

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

SCDS Name . . . . . . : SMS.SCDS1.SCDS
Management Class Name : INTERIM

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


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 29. Defining Aggregate Backup Attributes

The SCDS Name field specifies the name of the SCDS into which the management class is defined.

The # Versions field specifies the number of versions to be maintained.

The Retain Only Version field indicates how long the most recent backup version of an aggregate group is kept.

The Unit field specifies the unit of measure for the time period specified for the Retain Only Version field. This field cannot be blank if the Retain Only Version field is specified, except when Retain Only Version is NOLIMIT.

The Retain Extra Versions field indicates how long to keep backup versions of an aggregate group that precede the most recent version.

The Unit field specifies the unit of measure for the time period specified for the Retain Extra Versions field. This field cannot be blank if the Retain Extra Versions field is specified, except when Retain Extra Versions is NOLIMIT.

The Copy Serialization field specifies whether aggregate backup should continue if an enqueue failure is encountered.

The Abackup Copy Technique field specifies whether concurrent copy should be used in conjunction with aggregate group backup.

Assigning Management Classes

You can assign management classes through the management class ACS routine, by explicit specification or as part of an object class transition. For permanent system-managed data sets, the management class ACS routine is executed only if the storage class is valid. The management class ACS routine is not executed for temporary data sets.

An object's management class is determined when you store the object. The default management class for an object is defined in the collection to which the object belongs. (The default management class for an object collection is assigned by the management class ACS routine when the first object is stored in the collection.) This can be overridden if you specify a management class when you store the object. Then, the management class ACS routine is run to assign a management class to the object. The management class of an object can be changed with the OSREQ CHANGE macro or through class transition.

You can explicitly specify a management class on the following, but a management class determined by the management class ACS routine takes precedence:

- JCL DD statements
- TSO/E ALLOCATE command
- DFSMSdss COPY and RESTORE commands
- Access method services ALLOCATE, DEFINE, and IMPORT commands
- Dynamic allocation requests through ISPF/PDF data set allocation panels
- OSREQ STORE and OSREQ CHANGE macro requests

End users cannot override any of the attribute values that you assign to a data set or object through the management class ACS routine except for the expiration attributes, on the condition that the retention limit is not exceeded (for data sets or objects) or set to NOLIMIT (for data sets). See "Defining Management Class Expiration Attributes" on page 73 for information about the order of precedence for expiration attributes.

The syntax for specifying a management class on a JCL statement is:

```
MGMTCLAS=management-class-name
```

The syntax for specifying a management class on a TSO/E command is:

```
MGMTCLAS(management-class-name)
```

The syntax for specifying a management class on an access method services command is:

```
MGMTCLAS(-) or MANAGEMENTCLASS(management-class-name)
```

The syntax for specifying a management class on an OSREQ STORE or CHANGE macro is:

```
MGMTCLAS=management-class-area or MGMTCLAS=(management-class-area-pointer)
```

For information on determining management classes through ACS routines, see "Chapter 9. Defining ACS Routines" on page 137.

Defining Additional Management Classes

You can copy existing management classes and modify them to create new management classes by using the COPY line operator, which is explained in "Copying SMS Classes, Storage Groups, and Aggregate Groups" on page 209.

Chapter 6. Defining Storage Classes

Without SMS, you must place critical data sets on selected storage devices in order to improve performance. For example, if you have data sets that consistently require short response times, you place them on DASD volumes that have low I/O rates or that are connected to cache storage controllers. If you have data sets that require continuous availability, you can place them on dual copy, RAID volumes, peer-to-peer remote copy (PPRC), or extended remote copy (XRC).

SMS uses storage classes to separate data set performance objectives and availability from physical storage. It also provides attributes for the following:

- Sequential data striping for batch processing
- Allocation to a volume which supports concurrent copy, SnapShot, or FlashCopy™.
- VSAM record-level sharing between different systems with storage classes defined to use the coupling facility (CF).

This chapter describes storage classes and shows you how to define them through ISMF.

Duplication of Library Volsers: DUPT@SMS is an IBM reserved name and must not be defined in SCDS by the installation.

With STORCLAS=DUPT@SMS and DISP=OLD and VOL=SER=nnnnnn specified in the JCL, you can access an imported tape outside an automated library for input if you have another tape with the same volser in a system-managed library.

Understanding Storage Classes

A storage class is a list of storage objectives and requirements. Each storage class represents a list of services that are available to data sets and objects having similar access requirements. A storage class does not represent any physical storage, but rather provides the criteria that SMS uses in determining an appropriate location to place a data set or object.

In general, SMS attempts to select a location that meets or exceeds the specified objective, but it does not guarantee response time. If no location satisfies the performance objective, SMS attempts to find a location that most closely matches the specified objective for DASD data sets. If there is more than one device of similar characteristics to choose from, SMS selects the volume with the most available space for a majority of times.

Only you, as a storage administrator, can define or alter the storage classes for an SMS complex. End users can specify but cannot override storage classes assigned for a data set by the ACS routines.

Storage Classes for Data Sets

With the exception of tape data sets, a data set is system-managed if it has a storage class. When you assign a storage class to a system-managed DASD data set, SMS places the storage class name in both the BCS and the VVDS catalog entries of the data set.

© Copyright IBM Corp. 1986, 2002

To satisfy the availability requirements of critical data sets, SMS selects a volume that can provide data access even in the event of a single device failure. For example, this can be satisfied by a device that has dual copy active or that is an array DASD.

If you need to access data sets while they are being copied or backed up, you can place them on volumes that support concurrent copy, virtual concurrent copy, or FlashCopy.

With storage classes, you can also specify whether a data set is referenced primarily in the read or write mode.

For information on establishing dual copy volumes and enabling or disabling cache functions, see “Appendix B. SETCACHE Functions and Device Information” on page 303. In this section, you can also find information on IBM storage control units that support these features.

Tape data sets are not SMS-managed, even though you can assign them a storage class. Only tape volumes are SMS-managed. Therefore, none of the storage class attributes apply to tape data sets. Tape data sets do not have to be cataloged. If you do catalog a tape data set, the SMS information, such as data classes, storage classes, and management classes, is not saved in the catalog. You can direct tape data sets with a storage class to an SMS-managed tape volume via the storage group ACS routine.

Storage Classes for Objects

For OAM, storage classes separate performance objectives from physical storage and determines where the object resides in the storage hierarchy.

OAM uses storage classes to logically represent the level of service required by an object. This could result in placement of the object on one of the three types of physical storage in its object storage hierarchy:

- Direct access storage devices (DB2 object tables)
- Optical volumes
- Tape volumes

The default storage class for an object is defined in the catalog entry for the object collection to which the object belongs. If the default management class is not available from the catalog entry, OAM uses the collection entry in the DB2 table and updates the catalog entry. The default storage class is assigned by the storage class ACS routine when the first object is stored in the collection.

Note: If the catalog entry for the collection does not exist, but a DB2 table entry does, this is an error. If neither exist, both are created during a first time store to a new collection, using the storage class, management class, and storage group from the ACS routine.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for a more specific discussion of how to use storage classes with objects.

Planning Storage Classes

You should create a storage class for each level of service provided by the installation. Before you actually define storage classes, you need to identify the hardware available in your installation. The fastest response time and data sharing for VSAM data are obtained from the coupling facility (CF). The fastest response time for DASD data sets is obtained from cached devices, such as devices behind a 3990 Storage Control with cache active, an IBM RAMAC Virtual Array, or an IBM ESS. Object storage classes should reflect the hardware available in the object storage hierarchy.

Defining Storage Class Attributes

A storage class definition contains both descriptive and access-related information. To identify and refer to storage classes, you assign each one a unique name that contains from one to eight alphanumeric characters. Each storage class definition maintains an owner ID that identifies the storage administrator who originally created or last modified the storage class. Also, each storage class contains an optional 120 character description field for describing its contents.

This section discusses defining storage class attributes. See:

- "Listing SMS Classes, Aggregate Groups, Storage Groups, and Libraries Using ISMF" on page 195 for information on generating lists of storage classes
- "Altering Storage Classes" on page 203 for information on altering a storage class
- *z/OS DFSMS: Using the Interactive Storage Management Facility* for details on displaying storage class information

You can use ISMF to define storage class attributes by selecting option 3, Define a Storage Class, from the ISMF Primary Option Menu for Storage Administrators. Figure 30 illustrates the Storage Class Application Selection panel.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                      STORAGE CLASS APPLICATION SELECTION
Command ===>
To perform Storage Class Operations, Specify:
  CDS Name  . . . . . . . 'SMS.SCDS1.SCDS'
                                (1 to 44 character data set name or 'Active')
  Storage Class Name  . . SC01      (For Storage Class List, fully or
                                    partially specified or * for all)
Select one of the following options  :
  3  1. List          - Generate a list of Storage Classes
     2. Display       - Display a Storage Class
     3. Define        - Define a Storage Class
     4. Alter         - Alter a Storage Class
     5. Cache Display - Display Storage Classes/Cache Sets
If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

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

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

Figure 30. Defining a Storage Class

To define a storage class:

1. Specify a CDS NAME.

   The CDS NAME must be the name of an SCDS. ISMF primes the field with the last used name. The default is the quoted word 'ACTIVE', which represents the currently active configuration, but you cannot define storage classes to the 'ACTIVE' configuration.
2. Specify the STORAGE CLASS NAME of the storage class that you are defining. ISMF primes the field with the name last used.
3. Select option 3, DEFINE, and press ENTER.

   ISMF displays page 1 of the Storage Class Define panel, shown in Figure 31. On this panel, you select the storage class attributes.

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

SCDS Name . . . . . : USER8.TEST.SCDS
Storage Class Name  : SC1
To DEFINE Storage Class, Specify:
  Description ==> DEFINE STORAGE CLASS
              ==>
  Performance Objectives
   Direct Millisecond Response . . . .            (1 to 999 or blank)
   Direct Bias . . . . . . . . . . . .            (R, W or blank)
   Sequential Millisecond Response . .            (1 to 999 or blank)
   Sequential Bias . . . . . . . . . .            (R, W or blank)
   Initial Access Response Seconds . .            (0 to 9999 or blank)
   Sustained Data Rate (MB/sec)  . . .            (0 to 999 or blank)
  Availability . . . . . . . . . . . N            (C, P ,S or N)
  Accessibility  . . . . . . . . . . N            (C, P ,S or N)
   Backup  . . . . . . . . . . . . . .            (Y, N or Blank)
   Versioning  . . . . . . . . . . . .            (Y, N or Blank)

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 31. Defining Storage Class Attributes, Page 1 of 2

The SCDS Name and Storage Class Name fields are output fields that contain the SCDS and storage class names you specified in the Storage Class Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the storage class.

You can leave the Storage Class Define panel at any time without saving the storage class by issuing the CANCEL command.

The following sections describe the storage class attributes you can define.

Defining Performance Objectives

In the performance objectives fields of the Storage Class Define panel, you can request millisecond response (MSR) times and indicate the bias of both direct and sequential access data sets. All of the performance attributes are optional.

If you leave all MSR and bias fields blank (direct and sequential), SMS ignores device performance during volume selection. With the introduction of the IBM Enterprise Storage Server™ (ESS) and large cached controllers, the storage class MSR value is less significant except as noted below.

Tip: The MSR and bias values you specify in the storage class can be used to determine how buffers are to be allocated when system-managed buffering is used for VSAM applications. To do this, specify a value of `System` for the Record Access Bias attribute in the data class. This capability applies only to system-managed VSAM data sets allocated in extended format and accessed by batch applications. MSR values are also used to cache PDSE members and HFS files.

Defining Millisecond Response Time

The MSR serves two purposes in SMS. First, it is used as the performance objective for selecting candidate volumes for new data set placement. During a new data set allocation, SMS searches for a volume that meets or closely matches this objective. If no volume satisfies the objective, then SMS attempts to find a volume that comes closest to matching it. If more than one MSR is explicitly or implicitly specified, the storage class and associated device MSRs are averaged and compared.

Second, if the data is placed on a volume attached through an IBM 3990 Storage Control with cache, and cache is enabled for that volume, the MSR is used to determine if caching is mandatory, optional, or should be inhibited for the data set. This attribute does not apply to objects.

You can request SMS to ignore various device performances during volume selection by leaving all MSR and BIAS fields blank. This lets you spread data evenly across noncached and cache active devices.

Millisecond Response Time and Data Set Allocation: DASD can have different performance capabilities for direct access (random access, for example) and for sequential access applications. Its performance capabilities depend on whether you are reading data or writing data.

Each device type and model has a predetermined MSR capability for each condition. Additionally, if the device is attached to a cache capable control unit, the response capabilities are improved when caching is active. Therefore, each device is represented in Figure 32 on page 90 by eight MSR values:

- Uncached Performance
  - Direct Read MSR
  - Direct Write MSR
  - Sequential Read MSR
  - Sequential Write MSR
- Cache performance, DASD Fast Write performance, or both (if active)
  - Direct Read MSR
  - Direct Write MSR
  - Sequential Read MSR
  - Sequential Write MSR

If a device is cache capable, it must also have caching active at the time of allocation in order to be represented by the caching MSR values.

		UNCACHED VALUES[1]				CACHE VALUES				CACHE/DASD FAST WRITE				
		DIRECT		SEQUEN-TIAL		DIRECT		SEQUEN-TIAL		DIRECT		SEQUEN-TIAL		
DEVICE	CONTROLLER	RD.	WT.	RD.	WT.	RD.	WT.	RD.	WT.	RD.	WT.	RD.	WT.	IART[2]
3350 3375	3880-1[4]	30	30	30	30	30	30	30	30	30	30	30	30	0
3380	3880-3[4]	25	25	25	25	25	25	25	25	25	25	25	25	0
	3880-13[3]	25	25	25	25	14	25	14	25	14	25	14	25	0
	3880-23[3]	25	25	25	25	10	25	10	25	10	25	10	25	0
3380 3390-1 3990-2 3390-3	3990-2[4]	25	25	25	25	25	25	25	25	25	25	25	25	0
3380 3390-1 3390-2 3390-3	3990-3/6	25	25	25	25	10	25	10	25	10	6	10	6	0
3390-9	3990-2[4]	50	50	50	50	50	50	50	50	50	50	50	50	0
	3990-3/6	50	50	50	50	35	50	25	50	35	30	25	25	0
9345	9343-x0x[4]	25	25	25	25	25	25	25	25	25	25	25	25	0
	9343-xCx[3]	10	25	10	25	10	25	10	25	10	25	10	25	0
9395	9394	10	6	10	6	10	6	10	6	10	6	10	6	0
9392-1 9392-2 9392-3	3990-3/6	10	25	10	25	10	25	10	25	10	6	10	6	0
9393 9396		10	25	10	25	10	25	10	25	10	6	10	6	0
9397		4	4	4	4	4	4	4	4	4	4	4	4	0
3995-151[4]		600	620	230	250	600	620	230	250	600	620	230	250	10
3995-153[4]		575	590	200	215	575	590	200	215	575	590	200	215	10
2105		10	6	10	6	10	6	10	6	10	6	10	6	0

Notes:
1. Uncached Values of a device are always used an MSR of 999 is coded.
2. Initial access response time (IART) value = seconds; all other = milliseconds.
3. Does not support DASD FAST WRITE.
4. Does not support CACHE, does not support DASD FAST WRITE.

DA6S2017

Figure 32. MSR Capabilities

From the numbers included in Figure 32, you can observe that:

- The fastest write performance that can be currently achieved is 6 milliseconds.
- The fastest read performance that can be currently achieved is 10 milliseconds.
- When cache fast write is not active, write performance is the same as for an uncached device.

You can, for example, help direct allocation to a 3390-9 by specifying an appropriate MSR. A direct MSR of 35 milliseconds with a bias of read would favor a cache-active 3390-9.

Millisecond Response Time and Cache Management: At run time, when control unit caching is available, data is divided into three categories: must cache, may cache, and never cache.

Dynamic data set cache management is an SMS feature that permits expanded use of a storage controller that supports dynamic cache management enhancement (DCME) with cache, caching and DASD fast write features when these features are under-used or provides restricted use when they are overused. Only system-managed data sets that reside on volumes attached through a storage controller that supports DCME with cache are affected. Data sets not managed by SMS and data sets that reside on volumes attached through storage controls other than a storage controller that supports DCME with cache are not affected.

Dynamic cache management improves performance when a storage controller that supports DCME with cache is overloaded. It implements a cache management algorithm that optimizes the selection of data sets that are cache candidates. This enhancement also prevents over-commitment of nonvolatile storage. New I/O statistics by data set and SMS storage class can be collected and monitored by using the System Management Facility (SMF). Refer to *z/OS MVS System Management Facilities (SMF)* for details on the structure of SMF type 42 records.

As a function of dynamic data set cache management, each system-managed data set is assigned a cache usage attribute and a DASD fast write usage attribute for sequential and direct accessing modes. These usage attributes are based on the MSR and bias specifications found in the storage classes associated with the data set. These attributes are:

- Cache usage attribute:

  The data set might have one cache usage attribute for sequential accessing mode and a different attribute for direct accessing mode.

  Must Cache
  The respective MSR specification can be met only through the use of cache. BIAS=Read, Write or blank.

  As the name implies, this is data that must always be cached in order to meet the performance requirement. If you specify an MSR that is lower than any device can provide without caching, then

  – allocation places the data on a device closest to the requested MSR that has cache active (if one is available) and
  – the data is cached at execution time (when cache is active) because the MSR cannot be met by the uncached performance of the device.

  It is possible that as new devices are introduced, equivalent MSR might be obtainable without the use of cache. In that case, the data might be allocated to a noncached device, but the MSR is still achievable.

  May Cache
  The respective MSR specification can be met without the use of cache. BIAS=Read, Write or blank.

  Any data that is on a cache capable control unit is considered a candidate for *may cache*. At execution time, if the data is neither must cache nor never cache, dynamic cache management determines if and when to cache the data.

  Never Cache
  Never Cache data is data that is known to be cache unfriendly. An MSR of 999 indicates that data is never to be cached at execution time, even if it resides on a cache active device. SMS volume selection prefers a volume whose performance is equivalent to 25ms and has Cache inactive. See Table 4 on page 92.

- DASD fast write usage attribute:

  The data set might have one DASD fast write usage attribute for sequential accessing mode and a different attribute for direct accessing mode.

  Must DASD Fast Write
  The respective MSR specification cannot be met without the use of DASD fast write. BIAS=Write.

  May DASD Fast Write
  The respective MSR specification can be met without the use of DASD fast write. BIAS=Write, Read or blank.

  Never DASD Fast Write
  The respective MSR specification is 999.

Recommendation: Direct your IDCAMS query regarding the status of cache/DFW to the controller, since querying the device can lead to incorrect conclusions.

Defining Bias

Bias determines which volumes MSR performance numbers (read, write, or both) to consider during volume selection. If you specify a read (R) bias, a cache storage control should be available to allow caching. If you specify a write (W) bias, the DASD fast write feature of an IBM 3990 Storage Controller with cache should be available to allow the use of DASD fast write. If you do not specify a value for bias (blank), the MSR time determines whether caching or DASD fast write are used.

Once the data sets have been allocated, the MSR time determines whether caching or DASD fast write are used. You can inhibit caching or DASD fast write by specifying a MSR time of 999. If a bias is specified without an accompanying MSR, an MSR of 6ms for write bias or an MSR of 10ms for read bias is used for volume selection. If the MSR is blank, then the data set is May Cache and May DASD Fast Write.

Tip: Allocation does not fail when a storage control device does not have caching or DASD fast write available.

Table 4 can help you determine which setting (May, Must, Never) is set by DCME:

Table 4. D/T3990 SMS Cache Candidate Tokens for Sequential and Direct Requests

		Direct Cache Token		**Sequential Cache Token**	
Direct/Sequential MSR	**Dir/Seq Bias**	**Read**	**Write[3]**	**Read**	**Write[3]**
999	any	Never	Never	Never	Never
blank	blank	May	May[2]	May[1]	May
MSR < UCr[4] and MSR < UCw[5]	blank	Must	May[2]	Must	May
MSR < UCr and MSR >= UCw	blank	Must	May[2]	Must	May
MSR >= UCr and MSR < UCw	blank	May	May[2]	May[1]	May
MSR >= UCr and MSR >= UCw	blank	May	May[2]	May[1]	May
blank	Read	May	May[2]	May[1]	Never
MSR < UCr and MSR < UCw	Read	Must	May[2]	Must	Never
MSR < UCr and MSR >= UCw	Read	Must	May[2]	Must	Never
MSR >= UCr and MSR < UCw	Read	May	May[2]	May[1]	Never
MSR >= UCr and MSR >= UCw	Read	May	May[2]	May[1]	Never

Table 4. D/T3990 SMS Cache Candidate Tokens for Sequential and Direct Requests (continued)

		Direct Cache Token		Sequential Cache Token	
Direct/Sequential MSR	**Dir/Seq Bias**	**Read**	**Write[3]**	**Read**	**Write[3]**
blank	Write	May	May[2]	May[1]	May
MSR < UCr[4] and MSR < UCw[5]	Write	Must	Must	Must	Must
MSR < UCr and MSR >= UCw	Write	Must	May[2]	Must	May
MSR >= UCr and MSR < UCw	Write	May	Must	May[1]	Must
MSR >= UCr and MSR >= UCw	Write	May	May[2]	May[1]	May

Notes:

1. The 3990 Extended Platform results in any sequential read request to always be cached unless MSR 999 is specified.
2. The 3990 Extended Platform results in all direct write requests from VSAM or Media Manager to be treated as must write.
3. All write requests to a RAMAC device are treated as must write.
4. Uncached values read performance.
5. Uncached values write performance.

See Figure 32 on page 90 about performance characteristics of a particular IBM device.

Set the storage class to indicate the desired performance for a data set. SMS volume selection and DCME work together to provide the closest performance possible at the time of allocation and open.

Defining Initial Access Response Seconds

VTS (Virtual Tape Server) Cache Management: An initial access response time (IART) of 100 or greater means the volume has least preference in the cache. 0-99 means the volume has most preference.

Object Access: You can use the Initial Access Response Seconds field to specify the desired response time (in seconds) required for locating, mounting, and preparing a piece of media for data transfer. OAM uses this value to interpret the storage level, that is, to place an object at an appropriate level in the object storage hierarchy. For objects, both the IART and the sustained data rate (SDR) are applicable.

OAM uses IART as follows:

- If the IART value is 0, OAM writes to DASD.
- If the IART value is 1-9999, OAM selects removable media, either optical or tape. If the SDR is greater than or equal to 3, the primary copy of the object is stored on a tape volume. If the SDR for the object is less than 3, the primary copy of the object is stored on an optical disk volume.

The IART for an optical volume depends on its location. The time required for an operator to mount a shelf-resident optical volume is significantly longer than that for the automatic mounting of a library-resident optical volume. You can use up to four characters to specify a valid value of 0-9999, or leave the field blank. The default is blank and selects fixed DASD. However, an OAM request fails if it tries to use a storage class with a blank IART value.

Data Set Access: SMS allows the system resources manager (SRM) to select a DASD volume from the primary volume list if the IART value is 0 or unspecified. SRM volume selection is ideally suited for batch jobs.

You can allow SMS, instead of SRM, to select a DASD volume by assigning a storage class with a nonzero IART value. In this case, none of the eligible volumes is placed on the primary list; all of the eligible volumes are placed only on the secondary or tertiary list. SMS uses the randomizing technique, for example, so as not to always select the volume that has the most free space.

See “SMS Volume Selection for Data Set Allocation” on page 102 for further information on SMS volume selection.

Defining Sustained Data Rate

The Sustained Data Rate (SDR) field applies to physical sequential data sets, VSAM data sets, and objects.

Physical Sequential and VSAM Data Sets: The Sustained Data Rate (SDR) has an effect only for extended format data sets. Striping allows you to spread data across different DASD volumes and controllers. The number of stripes is the number of volumes on which the data set is initially allocated. Striped data sets must be system-managed and must be in an extended format.

When no volumes supporting striping are available, the data set is allocated as nonstriped with EXT=P specified in the data class; the allocation fails if EXT=R is specified in the data class.

Physical sequential data sets cannot be extended if any of the stripes cannot be extended. For VSAM data sets, each stripe can be extended to an available candidate volume if extensions fail on the current volume.

If you do not specify Guaranteed Space=Y in the storage class, the following is true:

- The system uses the SDR field in conjunction with the device transfer rate to derive the number of stripes that a striped data set can have (maximum of 16). The SDR field specifies the target throughput rate. Valid values for this attribute are blank or 0 to 999.
- For nonguaranteed space requests, if the SDR field is blank or 0, the target number of stripes is 1. If the value in the SDR field is greater than 1, it is divided by 4 for 3390s, or by 3 for 3380s to determine the stripe count. For example, for an SDR value of 24, a storage group of 3380s would have a stripe count of 8, and a storage group of 3390s would have a stripe count of 6. The volume must be able to satisfy the requested primary space.
- Depending on the number of eligible volumes in the candidate storage groups, the actual number of stripes allocated to a data set can be less than the derived one. As the stripe count is decreased, the primary space for each stripe increases, except for the guaranteed space requests.

If you specify Guaranteed Space = Y in the storage class, and for further information about guaranteed and nonguaranteed space allocation, see “Rules for Striping Volume Selection” on page 107.

See *z/OS DFSMS: Using Data Sets* for more information on using SMS to assign storage classes for striped data sets.

Object Access: OAM uses the SDR field to determine whether the primary copy of the object should be placed on optical or tape media. An SDR value of 3 or greater places the data on tape media, and a value of 2 or less places the data on optical media.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for further information.

Defining Availability

The Availability field is used to specify whether data set access should continue in the event of a single device failure. Storage classes with a blank Availability field default to NOPREF.

CONTINUOUS

Specify an availability of CONTINUOUS if you do not want a device failure to affect processing. Only duplexed and RAID volumes are eligible for this setting.

If CONTINUOUS availability is specified, data is placed on a device that can guarantee that it can still access the data in the event of a single device failure. This option can be met by

- A dual copy volume
- An array DASD

PREFERRED Array DASD volumes are preferred over nonduplexed volumes. Dual Copy volumes are not candidates for selection.

STANDARD If data sets do not require such a high level of availability, specify STANDARD availability, which represents normal storage needs.

Specify an availability of STANDARD to cause processing of a data set to stop after a device failure. Simplex volumes are preferred over array DASD. SMS selects only volumes that are *not* dual copy. This attribute does not apply to objects. Array DASD are acceptable candidates for both STANDARD and CONTINUOUS availability requests.

NOPREF Simplex and array DASD are equally considered for volume selection. NOPREF is the default. Dual copy volumes are not candidates for selection.

Defining Accessibility

The storage class accessibility attribute defines the function of the hardware supporting the point-in-time copy, using either concurrent copy, virtual concurrent copy, or FlashCopy. The point-in-time copy allows database management systems (DBMSs) to take what appears to be an instantaneous copy of data or a ″fast″ point-in-time copy. The copy can be for backup purposes (generally to tape) or for copying a database from one set of DASD volumes to another. The accessibility attribute allows you to direct allocation of new data sets to DASDs connected to an IBM 3990 Storage Control unit with cache that supports the concurrent copy function, IBM RAMAC Virtual Array (RVA) devices with the SnapShot copy support, DFSMSdss with the virtual concurrent copy, and the IBM ESS with the FlashCopy service.

The IBM Enterprise Storage Server (ESS) FlashCopy service provides the appearance of instantaneous replication of a range of track images. You can invoke

this service with DFSMSdss and use it to create copies for disaster recoveries, business intelligence applications, data in a test environment, and instantaneous checkpoints.

See *z/OS DFSMS Advanced Copy Services* and *z/OS DFSMS Software Support for the IBM Enterprise Storage Server* for further information on the ESS FlashCopy service.

When you specify accessibility attributes, you identify whether you want SMS to use *versioning* or *backup* devices for either concurrent copy or virtual concurrent copy.

Versioning Device
can create 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 ESS FlashCopy service.

Backup Device
can create 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 Use virtual concurrent copy. DFSMSdss uses the SnapShot feature of the RAMAC Virtual Array or the ESS FlashCopy service to create a point-in-time copy and then does I/O to create the backup to whatever target device you specified.

You can use a combination of values for the accessibility attribute and its subparameters, Versioning and Backup, to request the point-in-time copy. In the Accessibility field, you specify whether allocation to a point-in-time copy-capable volume is required (CONTINUOUS), preferred (CONTINUOUS PREFERRED), or discouraged (STANDARD). You then specify values for the Versioning and Backup subparameters to select which devices you want used for the copy.

The following defines your allocation request to a point-in-time copy-capable volume:

CONTINUOUS (C)
Only point-in-time copy volumes are selected.

CONTINUOUS PREFERRED (P)
Point-in-time copy volumes are preferred over non-point-in-time copy volumes.

STANDARD (S)
Non-point-in-time copy volumes are preferred over point-in-time copy volumes.

NOPREF (N)
Point-in-time copy capability is ignored during volume selection. This is the default.

Table 5 on page 97 identifies the values you specify based on your specific accessibility requirements:

Table 5. Combinations for Requesting Point-in-Time Copy Devices

If these are your choices for accessibility devices:			Specify the following values:
First Choice	**Second Choice**	**Third Choice**	

Versioning device	None	None	`Accessibility = C`
			`Versioning = Y`
			`Backup = N`

Method 1 backup device	None	None	`Accessibility = C`
			`Versioning = N`
			`Backup = Y`

Any versioning or backup device	None	None	`Accessibility = C`
			`Versioning = blank`
			`Backup = blank`

Any versioning or backup device	Any non-accessibility device	None	`Accessibility = P`
			`Versioning = blank`
			`Backup = blank`

Versioning or Method 2 backup device	Method 1 backup device	Any nonaccessibility device	`Accessibility = P`
			`Versioning = Y`
			`Backup = Y`

Versioning device	Any nonversioning device	None	`Accessibility = P`
			`Versioning = Y`
			`Backup = N`

Any nonaccessibility device	Any versioning or backup device	None	`Accessibility = S`
			`Versioning = blank`
			`Backup = blank`

Any device	None	None	`Accessibility = N`
			`Versioning = blank`
			`Backup = blank`

You can also use the data set alter application to alter a storage class so that data sets can be allocated on point-in-time copy-capable volumes.

Defining the Guaranteed Space Attribute

You can allocate space for single volume and multivolume data sets by specifying a storage class with the Guaranteed Space attribute. SMS fails the request when space is insufficient. The Guaranteed Space attribute does not apply to objects, but it does apply to both VSAM and non-VSAM-managed data sets.

For a multivolume system-managed data set, the system preallocates primary space on all the volumes. The first volume becomes the primary volume. The remaining volumes become candidate volumes with preallocated space. The system operates differently depending on what the user codes for DISP.

When the existing allocated space on the current volume becomes full with DISP=NEW or MOD, the system attempts to create secondary extents on the current volume. If not enough space is left on the current volume, the system uses the preallocated primary extent on the next volume. The system converts this next volume from a candidate volume to a primary volume.

When the allocated space (possibly including secondary amounts) on the current volume becomes full with DISP=OLD or SHR, the system switches to the next volume. The system allocates new space only on the last volume. The system converts each candidate volume to a primary volume.

For VSAM key-sequenced data sets with key ranges specified, KEYRANGES will be ignored if it is specified for the DEFINE CLUSTER command of IDCAMS.

For information about key ranges, see the description of the DEFINE CLUSTER command in *z/OS DFSMS Access Method Services*.

Preallocating Space for Multivolume Data Sets

The following JCL allocates a multivolume data set on SMS-selected volumes. The storage class DBLOG must have Guaranteed Space=Y. This example allocates 100 MB on each of five volumes. When all of the allocated space is used for the data set on one volume, the secondary space is allocated as required in extents on that volume. For example:

```
//DD1  DD  DSN=ENG.MULTFILE,DISP=(,KEEP),STORCLAS=DBLOG,
//         SPACE=(1,(100,25)),AVGREC=M,
//         UNIT=(3380,5)
```

1. After 100 MB is used on the first volume, 25 MB extents of secondary space is allocated on it until the extent limit is reached or the volume is full.
2. If more space is needed, 100 MB of primary space is used on the second volume. Then, more secondary space is allocated on that volume.
3. The same process is repeated on each volume, as shown in Figure 33 .

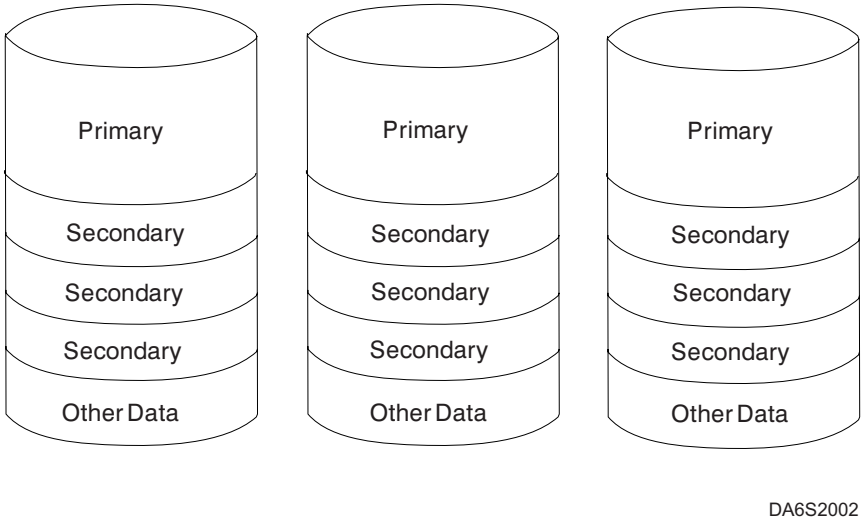


Figure 33. Allocation of Primary and Secondary Space for Multivolume Data Sets

Honoring Specific Volume Requests

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

Specifying volume serials 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 containing the volumes explicitly specified by the user.

- Ensure that all volumes specified by the user belong to the same storage group by directing an allocation with a Guaranteed Space attribute to all the storage groups in the installation.
- Ensure that the requested space is available on the volume 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.

With the IBM Enterprise Storage Server (ESS), the guaranteed space attribute of a storage class with specific volume serials is no longer required for data sets other than those that need to be separated (for example, DB2 online logs and BSDS) or that must reside on specific volumes because of their naming conventions (for example, VSAM RLS 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 the purpose of availability.

The ESS capabilities of multiple allegiance and parallel access volumes (PAV), along with its bandwidth and caching algorithms, make it unnecessary to separate data sets 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 once 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.

The ESS can also execute multiple concurrent accesses to a single volume from a single host or PAV. 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.

Defining Guaranteed Synchronous Write

You can use the Guaranteed Synchronous Write attribute to indicate whether your system should return from a BSAM CHECK (or WAIT) issued for a WRITE against a PDSE member before (unsynchronized) or after (synchronized) the data has actually been written to DASD. This attribute does not apply to objects.

Specify a Y for synchronized write or an N for no synchronization.

Defining Use of the Coupling Facility for VSAM Record-Level Sharing

This section describes how to associate a storage class with a CF cache set defined in the base configuration, thereby making any data set associated with the storage class eligible for VSAM record-level sharing. It also describes how to assign a weight value to the data, so as to indicate its relative importance.

1. Press the DOWN key or use the DOWN command to view the second page of the Storage Class Define panel, shown in Figure 34 on page 100:

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

SCDS Name . . . . . : USER8.TEST.SCDS
Storage Class Name  : SC1

To DEFINE Storage Class, Specify:

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









Use ENTER to Perform Verification; Use UP Command to View previous Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 34. Defining Storage Class Attributes, Page 2 of 2

2. Enter the name of a CF cache set defined in the base configuration.

   When you specify a cache set name, any data set associated with this storage class becomes eligible for record-level sharing using the CF. CACHE SET NAME maps the storage class to a CF cache set defined in the SMS base configuration, for which CF cache structures have been defined.

   In a JES3 environment, be careful to define cache set names only in those SMS storage classes that are used by data sets opened for VSAM RLS processing. When you define a cache set name in a storage class, any job accessing a data set associated with that storage class is scheduled on a VSAM RLS-capable system. If all storage classes have cache set names defined for them, then all jobs accessing SMS-managed data sets are scheduled to VSAM RLS-capable systems. This could cause a workload imbalance between those systems and down-level systems.

3. Specify a weight attribute indicating the data's relative importance in the CF DIRECT WEIGHT or the CF SEQUENTIAL WEIGHT fields. Use the CF DIRECT WEIGHT field for direct data; use the CF SEQUENTIAL WEIGHT field for sequential data. The default is a weight value of 6.

   Restriction: DFSMS only supports the default value. All data is assigned a weight value of 6 regardless of the value you specify.

   See "Chapter 14. Administering VSAM Record-Level Sharing" on page 235 for more information about administering VSAM record-level sharing.

Assigning Storage Classes

With the exception of tape data sets, a data set is SMS-managed if it is assigned a storage class. You can assign storage classes either through the storage class ACS routine or by explicit specification. If you do not specify an explicit storage class when you store an object, the object is assigned the storage class that is defined in the collection to which the object belongs. The default storage class for an object collection is assigned by the ACS routine when the first object is stored in that collection. If the storage class ACS routine determines a storage class, it takes precedence over one that is explicitly specified by any of the following:

- JCL DD statements
- TSO/E ALLOCATE command
- DFSMSdss COPY and RESTORE commands
- Access method services ALLOCATE, DEFINE, and IMPORT commands
- Dynamic allocation requests, such as with ISPF/PDF data set allocation panels
- OSREQ STORE and CHANGE requests

ACS routines are required for Distributed FileManager/MVS-created data sets to ensure that they are SMS-managed. These data sets must be SMS-managed. If a remote application attempts to create a local data set in non-SMS-managed storage, Distributed FileManager/MVS refuses to honor the request because it only creates SMS-managed data sets. Distributed FileManager/MVS does, however, support the access of non-SMS-managed data sets.

The syntax for specifying a storage class on a JCL statement is

```
STORCLAS=storage-class-name
```

The syntax for specifying a storage class on a TSO/E command is:

```
STORCLAS(storage-class-name)
```

The syntax for specifying a storage class on an access method services command is:

```
STORCLAS(-)
```

The syntax for specifying a storage class on an OSREQ STORE or CHANGE macro is:

```
STORCLAS=storage-class-area or STORCLAS=(storage-class-area-pointer)
```

For information on determining storage classes with ACS routines, see “Chapter 9. Defining ACS Routines” on page 137.

Defining Additional Storage Classes

You can copy existing storage classes and modify them to create new storage classes by using the COPY line operator, which is explained in “Copying SMS Classes, Storage Groups, and Aggregate Groups” on page 209.

SMS Volume Selection for Data Set Allocation

When a data set is allocated, SMS follows a sequence of verification steps to determine placement on a volume. This section describes how SMS selects a volume based on the requirements of a data set. It also explains both conventional volume selection and striping volume selection. Finally, it discusses why volume selection might fail.

Before the volume selection process begins, SMS categorizes each volume in a storage group into one of four lists from which volumes are selected for data set allocation:

Primary All the volumes in all the specified storage groups are candidates for the first, or primary list. The primary list consists of online volumes that meet all the specified criteria in the storage class and data class, are below threshold, and whose volume status and storage group status are enabled. All volumes on this list are considered equally qualified to satisfy the data set allocation request. Volume selection starts from this list.

Secondary Volumes that do not meet all the criteria for the primary volume list are placed on the secondary list. If there are no primary volumes, SMS selects from the secondary volumes.

Tertiary Volumes are marked for the tertiary list if the number of volumes in the storage group is less than the number of volumes requested. If there are no secondary volumes available, SMS selects from the tertiary candidates.

Rejected Volumes that do not meet the *required* specifications (ACCESSIBILITY = CONTINUOUS, AVAILABILITY = STANDARD or CONTINUOUS, ENABLED or QUIESCED, ONLINE...) are marked rejected and are not candidates for selection.

After the system selects the primary space allocation volume, that volume's associated storage group is used to select any remaining volumes requested for the data set. If you specify an extend storage group, the data may be extended to the specified extend storage group. For information on how to specify an extend storage group using the Extend SG Name parameter, see "Defining a Pool Storage Group" on page 51.

Restriction: SMS does not check or verify DASDVOL authorization for data set allocations on SMS-managed volumes. By adding a volume to SMS you are removing it from any active DASDVOL authorizations because DASDVOL access is not verified on SMS-managed volumes. See *z/OS Security Server RACF Security Administrator's Guide* for further information on DASDVOL authority.

Conventional Volume Selection

Conventional volume selection is used for all nonstriped data sets, as well as for data sets allocated with an SDR or zero or blank.

Table 6 represents the order that SMS uses to determine the preference sequence of volume selection for allocation of a data set.

Table 6. Volume Selection Preferences

Criteria	Preferences	Preference Value[1]
DATA SET SEPARATION	Volume does not reside in the same physical control unit that has allocated a data set from which this data set should be separated, as specified in the data set separation profile.	4096
VOLUME COUNT	Volume resides in a storage group that has enough eligible volumes to satisfy the requested VOLUME COUNT.	2048
HIGH THRESHOLD	Volume has sufficient space for the allocation amount without exceeding the storage group HIGH THRESHOLD value.	1024
SMS STATUS	Volume and its associated storage group SMS status are ENABLED.	512
END-OF-VOLUME EXTEND	This is an end-of-volume extend, where an extend storage group is specified, and the volume does not reside in the specified extend storage group.	256
NONOVERFLOW	Volume resides in a nonoverflow storage group.	128
IART	Volume is mountable, and an IART value greater than zero was specified in the storage class.	64
SNAPSHOT	Volume is in the same SnapShot-capable controller as the data set, and this is a snap request.	32
ACCESSIBILITY	Volume resides in a control unit that supports ACCESSIBILITY, and the storage class ACCESSIBILITY value is PREFERRED.	16
	Volume resides in a control unit that does not support ACCESSIBILITY, and the storage class ACCESSIBILITY value is STANDARD.	
AVAILABILITY	Volume resides in a control unit that supports AVAILABILITY and the storage class AVAILABILITY value is PREFERRED.	8
	Volume resides in a control unit that does not support AVAILABILITY and the storage class AVAILABILITY value is STANDARD.	
EXTENDED FORMAT	Volume resides in a control unit that supports EXTENDED FORMAT and the data class IF EXT value is PREFERRED.	4
MILLESECOND RESPONSE (MSR)	Volume provides the requested response time that is specified or defaulted in the storage class DIRECT MSR or SEQUENTIAL MSR.	2
	Volume provides a faster response time than what is specified or defaulted in the storage class DIRECT MSR or SEQUENTIAL MSR.	1

1. If a criterion is not requested or not met, its preference value is 0.

The volume selection criteria in Table 6 are weighted by preference. SMS selects the volume with the highest cumulative preference value. Figure 35 on page 104 provides an example of this process.

```
Volume A meets the following volume selection preference criteria:

      Volume does not have sufficient space for the allocation
      amount without exceeding the Storage Group HIGH THRESHOLD value      0

      Volume and its associated Storage Group SMS status are ENABLED     512

      Volume resides in a nonoverflow Storage Group                      128

      Volume resides in a control unit that supports ACCESSIBILITY,
      and the Storage Class ACCESSIBILITY value is PREFERRED              16

Total preference value for Volume A:                                     656


Volume B meets the following volume selection preference criteria:

      Volume has sufficient space for the allocation amount without
      exceeding the Storage Group HIGH THRESHOLD value                  1024

      Volume and its associated Storage Group SMS status are ENABLED     512

      Volume does not reside in a nonoverflow Storage Group                0

      Volume resides in a control unit that does not support
      ACCESSIBILITY, and the Storage Class ACCESSIBILITY value
      is PREFERRED                                                         0

Total preference value for Volume B:                                    1536

Because Volume B has the higher total preference value, SMS selects Volume B.
```

Figure 35. Example of Evaluation Process for Volume Selection

Primary List

All the volumes in all of the specified storage groups are candidates for the first, or primary list. The primary list consists of those volumes that fit *all* the storage class and data class requirements, are online, enabled and below threshold. All volumes on this list are considered equally qualified to satisfy the request data set allocation request.

A choice is made from the candidates on this list, preferring volumes that are not already allocated to a job or subsystem and that have the least I/O delay as determined by the system resource manager (SRM). It is possible that no volumes will appear on the primary list, if no devices in the storage group(s) meet all of the primary list criteria. It is also possible that even though there are devices on this primary list, the data set cannot be successfully allocated to any of these devices (i.e., there is not enough space available when the actual allocation request is made.)

For example, if you specify a 25 MSR in your storage class, volumes that are close to an MSR of 25 would be POOL1 volumes. POOL1 volumes are those volumes that meet your MSR requirement by a certain percentage. Each MSR percentage represents a range of MSRs. All volumes that fall within an MSR percentage are considered equal in performance. Volumes that provide a faster MSR by a larger percentage than POOL1 volumes are considered POOL2 volumes. POOL2 volumes include POOL1 volumes plus the volumes in the next higher MSR percentage. POOL1 volumes are preferred over POOL2 volumes.

Primary volumes include online POOL1 and POOL2 volumes that can meet the storage group criteria listed in Table 6 on page 103, in addition to guaranteed space requirement.

Volume selection from the primary list can result in skewing in extreme circumstances. These circumstances include the occasions when a VTOC index is disabled or when a new volume is added to SMS and selected most of the time. SMS uses the randomizing technique to avoid favoring one volume over others, such as an empty volume, a volume with a disabled index, or a volume under the allocation threshold. You can force SMS to always use the randomizing technique by specifying a nonzero IART value. This causes all fixed DASD volumes to be placed on the secondary list, which uses randomization between similar volumes.

Secondary List

The secondary list contains all volumes that can be used for data set allocation, even if they do not meet all of the storage class criteria. The secondary list also includes all volumes from the primary list.

If a data set cannot be allocated on a primary volume, SMS will continue to evaluate volumes on the secondary list until allocation is successful. SMS evaluates volumes in the secondary list based upon volume characteristics, performance, available space and system accessability. See Table 6 on page 103 for more information about volume selection preferences.

Tertiary List

When a storage group does not contain enough volumes to satisfy the volume count, all volumes in the storage group are flagged as tertiary. Tertiary volumes are only selected when there are no primary or secondary volumes and the request is for a non-VSAM, non-GUARANTEED SPACE request. The concept of tertiary volumes does not apply to VSAM data sets. In other words, for all VSAM non-GUARANTEED SPACE requests, the volume count does not play a role in determining which storage group is selected.

Space Constraint Relief

If SMS cannot allocate the data set to any of the volumes, SMS will perform space constraint relief (if specified in the data class) and repeat the selection process.

Reasons a Volume is Not Placed on the Primary List

A volume is not placed on the primary list if it does not meet all of the criteria listed in Table 6 on page 103, or for one of the following reasons:

- The volume is rejected. See “Possible Reasons for Volume Selection Failure” on page 109 for more information.
- The VTOC index is broken or disabled, resulting in SMS not being updated with space statistics from the Common VTOC Access Facility (CVAF). This can also happen if OEM products bypass CVAF, the component that notifies SMS of space changes. SMS not being updated with space statistics can also result in a volume being overused.
- The volume was placed on the tertiary list because the number of volumes in the storage group was less than the number of requested volumes. (See “Tertiary List” for more information on the tertiary list.)

  For example, this happens when a tape request of five volumes causes a tape mount management buffer storage group containing only two volumes to be marked tertiary, while an overflow storage group or a quiesced storage group containing the requested five volumes is marked secondary. The overflow volumes and quiesced volumes on the secondary list are preferred over the tape mount management volumes on the tertiary list.

SMS will attempt to allocate space using the overflow volumes first, then the quiesced volumes, and lastly the tape mount management buffer volumes.

- The controller was IMLed while online to MVS. This can result in the MVS device control blocks not reflecting the current state of the volume. The device or devices should be varied offline and back online to update the MVS control block status.
- RAID devices are eliminated from the primary list when STANDARD is specified. Use NOPREF, the AVAILABILITY default, to allow both SIMPLEX and RAID devices to be placed on the primary list.

For more information to help you determine why a particular volume is selected or not selected, see Table 6 on page 103.

Tuning Considerations

Be careful when tuning your storage class ACS routines to match the mixture of volume capabilities assigned by your storage group ACS routines. For example, if you have both cached and noncached 3390-3 in the assigned storage groups and your ACS routine assigned the same storage class (MSR=25) to all data sets, the noncached volumes are preferred over the cached volumes. This results in the noncached volumes being overused and the cache-active volumes are not selected until all the noncached volumes are above high threshold. If an MSR=2 is specified, then the cached volumes are overused.

This also holds true for other attributes. If you always assign ACCESSIBILITY=PREFERRED, then concurrent copy volumes are overused.

Spreading Allocations Across Multiple Volumes

You might want to separate allocations and spread them across multiple volumes to reduce I/O contention and improve performance. After analyzing your data, you might find that certain data may not need some of the advanced storage features in your environment.

If you want to spread conventional allocations across volumes of different features, use one or all of the following suggested parameter settings in the assigned storage and data classes:

Storage Class Parameters

- MSR = blank
- BIAS = blank
- ACCESSIBILITY = NOPREF
- AVAILABILITY = NOPREF
- GUARANTEED SPACE = N

Data Class Parameter

- if EXT = blank

Striping Volume Selection

Striping volume selection is entered only in the following situations:

- During initial allocation:

  If the data class specifies an extended format as either ’required’ or ’preferred’ and the SDR value in the storage class is nonzero. If the SDR value is zero, a nonstriped data set in extended format is allocated, and conventional volume selection is used.

  Tip: A nonstriped data set in extended format may refer to a *single-striped data set*.

- During restore/recall processing:

  If the data set was a multistripe data set when it was migrated or backed-up. If the data set was single-stripe when migrated or backed-up, it follows the conventional volume selection path.

Volumes are classified as primary and secondary, and primary volumes are preferred over secondary volumes. A single primary volume is randomly selected for each unique controller. All other eligible volumes behind the same controller are secondary. Secondary volumes are randomly selected if initial allocation on the primary volume is unsuccessful. There is no preference in different controller models as long as the controller supports striping.

SMS randomly selects a storage group from a list of storage groups that contain at least as many unique controllers as the requested stripe count. If there are no storage groups that meet this criterion, the storage group with the largest number of unique controllers is selected (randomly if there are more than one).

Volumes that meet the requested MSR are preferred over volumes that do not meet the requested performance. In DFSMS/MVS 1.1, a volume is considered to meet the requested performance if the volumes meets or exceeds the request. In DFSMS/MVS 1.2 and later, a volume meets the requested performance if the volume's performance is within a predetermined range of the requested MSR.

The throughput of striped data sets is gated by the slowest device if the striped set includes devices of varying data delivery capabilities.

Rules for Striping Volume Selection

For both guaranteed and nonguaranteed space allocations, the following key rules apply for striping volume selection:

- Storage groups containing mixed device types are not considered.
- Eligible volumes must remain below the ″high threshold″ defined in the storage group after the stripe is allocated. This is different from nonstriped allocations where no volume is rejected because of space, although they can be given a lower priority.
- Quiesced and enabled volumes are treated identically for striped allocations, unlike nonstriped allocations where enabled volumes are preferred over quiesced volumes.
- Target volumes
  - An SDR of 0 results in a target stripe count of 1 for both guaranteed and nonguaranteed space requests, and conventional volume selection occurs. A nonzero SDR that is 1 when divided by the device value results in a target stripe count of 1, but striping volume selection is used instead of conventional volume selection.
  - For nonguaranteed space, the number of target volumes is computed by dividing the SDR that is specified in the storage class by a value of 3 for 3380 devices, by a value of 4 for 3390 devices, and rounding up the result, if required. For example, an SDR of 18 results in a target stripe count of 6 on a 3380 device and a target stripe count of 5 (after rounding up) on a 3390 device.
  - For guaranteed space, if the SDR value is 1 or greater, it is ignored; in which case, the target number of stripes is the greater of either the volume count that is specified or the number of specified volume serial numbers. The number of stripes is never greater than 16. All specified volumes must be in

the same SMS storage group for a guaranteed space request. SMS assumes that the amount of space that the user wants is the target number of stripes times the specified primary space.

- All temporary data sets with a volume count greater than one are allocated as nonstriped.
- The volume must be able to satisfy the primary space requested by the number of stripes and not exceed the high threshold specified in the storage group.
- The maximum number of stripes (volumes) is 16.
- The maximum number of extents is five for each space allocation.
- The maximum number of extents per stripe is 123. For VSAM data sets, the maximum number of extents:
  - Per volume is 123
  - Per stripe is 255
  - Per component is 4080
- The minimum allocation is one track per stripe.
- The maximum allocation can exceed the 64K track limit and is limited by the amount of free space below the high threshold or extent constraints mentioned above for primary space allocations.
- All stripes must be able to satisfy the secondary space allocation (divided by the number of stripes) during extend processing or the allocation fails. Secondary space amount is divided by the number of stripes for both guaranteed and nonguaranteed space requests. OPEN, CLOSE, and EOV perform this calculation.
- Non-VSAM multistriped data sets cannot be extended to additional volumes. Striped VSAM data sets can be extended to additional volumes.
- Volume fragmentation information is not available to SMS at volume selection time. An allocation failure by DADSM because of fragmentation results in striping volume reselection.
- Primary space
  - For nonguaranteed space, the volume must be able to satisfy the primary space that is requested divided by the number of stripes. For example, if primary space is 15 MB and the number of stripes is three, the volume must be able to satisfy an allocation of 5 MB and not exceed the high threshold as specified in the storage group.
  - For guaranteed space, for requests that contain specified volume serial numbers, each stripe must be able to satisfy the requested primary space (15 MB in above example).
- Secondary space amount is divided by the number of stripes and rounded up for each volume.

The following is true for guaranteed space allocations:

- If you explicitly specify volume serial numbers with guaranteed space and the target number of stripes is equal to the number of volume serial numbers that you specify, SMS must allocate the primary space requested on each of these volumes. If this is not possible, the allocation fails.
- If you do not specify any volume serial numbers, then the target number of stripes is equal to the volume count. SMS tries to select the same number of volumes, but settles for less if this number is unavailable. If fewer stripes are allocated for the non-VSAM data sets, SMS increases the allocation per volume to compensate for the fewer stripes. For VSAM data sets, the primary space requested will be allocated on each stripe regardless of whether the target number of stripes is acquired or not.

- For VSAM data sets, if the number of guaranteed space volumes exceeds sixteen (which is the maximum number of stripes for a VSAM data component), the number of guaranteed space volumes will be reduced to sixteen with the remaining volumes becoming candidates for secondary space.
- If the target number of stripes is higher than the number of volume serial numbers you specify, SMS must select all the specified volumes plus additional ones. These nonspecific volumes are not mandatory and if none are available, SMS allocates the primary quantity on each of the specified volumes.

Recommendation: Ensure that you have a sufficient number of volumes behind each controller in the storage group to reduce volume overuse. An example of a problem would be if a storage group contains eight controllers and the average stripe count is four, each controller is selected approximately 50 percent of the time. If one controller contained only two volumes, each of the two volumes is selected for approximately 25 percent of all new striped allocations.

Requirements for Data Set Striping

The following lists the requirements for data set striping.

- Volumes must be behind one of the following controllers:
  - Controllers that are ESCON-attached and support concurrent copy
  - 3990-6 controllers
  - 3990-3 controllers which are Extended Platform and ESCON-attached
  - 3990-3 controllers which have the RAMAC support-level microcode
  - 9394 controller
  - 9343 controller with cache
  - IBM RAMAC Virtual Array
  - IBM Enterprise Storage Server
- Volumes must be ENABLED or QUIESCED and varied ONLINE.

Possible Reasons for Volume Selection Failure

During the process of volume selection, volumes can be rejected for any one of many different reasons. Possible reasons for rejection include:

- The volume is not online to MVS.
- The SMS volume or storage group status is DISNEW, DISALL, or NOTCON.
- The volume was not initialized as being SMS-managed.
- The volume does not contain enough space to satisfy the primary space requested (this is determined by DADSM).
- The volume control data sets (VTOC) does not contain enough space to accommodate the FMT1 DSCB for the allocation request, as determined by DADSM.
- The volume was selected and DADSM returned an unsuccessful return/reason code for the allocation request. When DADSM fails to allocate the data set on a volume due to insufficient space on the volume, SMS tries to allocate the data set on a different volume. Other failures in DADSM can result in allocation failures.
- The volume does not meet the availability requirement specified in the storage class.
- The volume does not support concurrent copy, SnapShot, or FlashCopy and the storage class specifies accessibility continuous (Required).
- The DASD controller does not support extended format and the data class specifies 'If EXT=R.'
- The volume is mountable DASD when an IART of zero or blank was specified in the storage class.

- The volume was listed on the provided exclude list or was not listed in the provided include list during the DFSMSdss COPY or RESTORE operations.
- The volume has an incorrect unit control block (UCB) type.
- The volume was not specified for a specific guaranteed space request.
- The volume belongs to a storage group and the storage group does not contain enough volumes to satisfy the volume count of a guaranteed space request. All volumes in the storage group are rejected.
- The volume is too fragmented to contain the primary space extent.
- During extend (EOV) processing, volume selection rejects volumes whose device types do not match the device type of the volume that the data set currently resides on.
- The controller was IMLed while online to MVS. This can result in the MVS device control blocks not reflecting the current state of the volume. The device or devices should be varied offline and back online to update the MVS control block status. SMS can reject the device because of incorrect MVS status. ISMF reports the correct status because ISMF queries the device/controller directly. To not impact I/O performance, SMS uses the status in the MVS device control blocks for all volumes in a storage group.
- The volume is not assigned to any of the storage groups selected by the storage group ACS routine. This results in the volume not being included on the candidate volume list, which has the same effect as the volume being rejected.

Note: During the volume selection process, many volumes can be rejected because they do not possess the right attributes. With few exceptions, volume selection fails only after allocation has been attempted without success on each of the remaining volumes.

Chapter 7. Defining Data Classes

SMS simplifies data set allocations by introducing data classes. This chapter describes SMS data classes and explains how you can define data classes with the ISMF data class application.

Understanding Data Classes

A data class is a list of data set allocation attributes and their values. You cannot assign a data class to an object; however, data class may be used for allocation of a scratch tape to be used to write objects. When end users allocate a data set and refer to a data class either explicitly (for example, through JCL) or implicitly (through ACS routines), SMS allocates the data set using the attribute values of its associated data class. For data class attributes, explicit specifications by end users override any parameters derived from the data class ACS routine. If SMS is active, a data class can be assigned to any data set. For non-SMS-managed DASD data sets, the system uses the allocation attribute values of the data class, but it does not save the data class name. For tape data sets, only the expiration and retention values are applied.

Not all attributes apply to every data set organization. When SMS allocates a data set, it uses only those data class attributes that have meaning for the given data set organization. SMS saves the data class name for each SMS-managed data set. The actual data class definitions reside in the SCDS. If you alter a data class definition, SMS applies the changes to any new data sets that use the data class after you activate the changed configuration. However, SMS does not retroactively apply your changes to previously allocated data sets.

A data class definition contains identification and allocation information. To identify and refer to your data classes, you assign each one a unique name that contains from one to eight alphanumeric characters. Each data class maintains an owner ID that identifies the storage administrator who originally created or last modified the data class. The owner ID can be viewed on the Data Class List panel. Also, each data class contains an optional 120 character description field for describing its contents.

The data class allocation attributes match the keywords that you use to allocate data sets. The attributes contain space-related, access-related, and organizational information that you typically find on JCL DD statements, TSO/E ALLOCATE commands, access method services DEFINE commands, dynamic allocation requests, and ISPF/PDF panels.

Planning Data Classes

You should create a data class based on service level agreements. For example, all data sets having a low-level qualifier of LIST, LISTING, OUTLIST, or LINKLIST probably belong to the same data class, because they are typical work data sets having similar allocation characteristics.

Before you actually define any data classes, gather information about the common types of data sets in your installation. You also need to determine if only the data class ACS routine can assign data classes to data sets, if end users can assign data classes to data sets, or if you want a combination of these two policies. If you intend to have only the data class ACS routine assign data classes, you need to

© Copyright IBM Corp. 1986, 2002

develop methods to identify the data in your installation. However, if you allow only the data class ACS routine to assign data classes, you should be aware that users need to override some data class attributes. For example, you should probably not code one data class for each possible amount of space that users need. Finally, you need to identify the space requirements for some commonly used data sets.

By gathering useful installation information, you can relieve your end users from specifying all of the allocation attributes on allocation requests. They can use the data class that most closely matches their needs and explicitly change the few attributes that are unique to their data sets.

You can create a data class with no attributes specified and use it to handle system-managed data sets that are allocated (but never opened) without all DCB attributes being specified. These are called ″empty data sets.″ This might be a common situation with batch systems where the DSORG, for example, is specified in the program but not all allocated data sets are used for every run of the batch stream. The empty data sets then occupy space on DASD and cannot be migrated by DFSMShsm since they do not have a specified DSORG. To solve this, check in your data class ACS routine for allocations that do not specify a DSORG or a data class and assign them the blank data class. This causes the DSORG for the data set to default to physical sequential (PS), so that DFSMShsm can migrate the data set.

A data set that has been opened for output and closed without writing anything is called a ″null data set.″

Defining Data Class Attributes

You can use ISMF to define your data classes by selecting option 4, DATA CLASS, from the ISMF Primary Option Menu for Storage Administrators. Figure 36 illustrates the Data Class Application Selection panel.

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


To perform Data Class Operations, Specify:
  CDS Name . . . . . . 'SMS.SCDS1.SCDS'
                              (1 to 44 character data set name or 'Active' )
  Data Class Name  . . DATAV     (For Data Class List, fully or partially
                                 specified or * for all)

Select one of the following options  :
  3  1. List    - Generate a list of Data Classes
     2. Display - Display a Data Class
     3. Define  - Define a Data Class
     4. Alter   - Alter a Data Class

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria


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

Figure 36. Defining a Data Class

Add a CDS Name and a Data Class Name on the panel and select option 3, Define. The CDS Name must be the name of an SCDS. ISMF primes the CDS Name field with the name last used. The default is the quoted word 'ACTIVE', which represents the currently active configuration, but you cannot define or alter data classes to the 'ACTIVE' configuration.

In the Data Class Name field, you must specify the name of the data class that you are defining. ISMF primes the field with the last used name.

Defining Record and Space Attributes for Data Class

Select option 3, DEFINE, and press Enter to see page 1 of the Data Class Define panel shown in Figure 37. You can leave any of the pages of the Data Class Define panel at any time without saving the data class by issuing the CANCEL command.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                              DATA CLASS DEFINE                  Page 1 of 4
Command ===>

SCDS Name  . . : Y421252.MYSCDS
Data Class Name: DCA11

To DEFINE Data Class, Specify:
  Description ==>
              ==>
  Recfm . . . . . . . . . VB            (any valid RECFM combination or blank)
  Lrecl . . . . . . . . . 255           (1 to 32761 or blank)
  Space Avgrec  . . . . . U             (U, K, M or blank)
        Avg Value . . . . 255           (0 to 65535 or blank)
        Primary . . . . . 5000          (0 to 999999 or blank)
        Secondary . . . . 5000          (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 59 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 37. Defining Data Class Record and Space Attributes

SCDS Name and Data Class Name are output fields that contain the SCDS and data class names you specified in the Data Class Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the data class. All of the remaining fields are optional and have a default value of blanks.

A blank indicates that no value is assigned to a parameter, and end users need to specify the value explicitly if it is required and they intend to use the data class. Otherwise, their jobs might fail. When users override some data class attributes, they should ensure that the remaining attributes in the data class do not conflict with the explicitly specified attributes.

You can specify the following attributes on page 1 of the Define Data Class panel:

Recfm
specifies the data set record format. You can specify one of the following, with the option of appending either an A (ISO/ANSI control character) or an M (Machine control character):

U Undefined
V Variable
VS Variable spanned
VB Variable blocked
VBS Variable blocked spanned
F Fixed
FS Fixed standard
FB Fixed blocked
FBS Fixed blocked standard

If you specify Recfm, you cannot specify the Recorg attribute.

LRECL
specifies the logical record length in bytes. If you leave the Recorg field blank, you can specify a LRECL value from 1-32760 (or leave it blank). If you choose a Recorg value of KS, ES, or RR, you can specify a LRECL value from 1-32761 (or leave it blank). If the Recorg value is KS, the LRECL value must be greater than or equal to the value you specify for the Keylen attribute. Also if the Recorg value is KS and you specify values for both the Keylen and Keyoff attributes, then the LRECL value must be greater than or equal to the sum of the Keylen and Keyoff attributes.

If the Recorg value is LS, the LRECL attribute is ignored and the CIsize value is 4096.

Space
Specifies a request for space in records, eliminating track and cylinder space requests.

Avgrec
specifies a scaling factor for primary and secondary record allocations. It can have the following factors:

U Multiplies the allocation quantity by 1
K Multiplies the allocation quantity by 1024
M Multiplies the allocation quantity by 1048576

Avg Value
specifies the average length of each record.

Primary
specifies the primary allocation quantity of records as multiplied by the scaling factor (Avgrec).

Secondary
specifies the secondary allocation quantity of records as multiplied by the scaling factor (Avgrec).

Directory
specifies the number of directory blocks for a PDS. It is valid only when the following fields are blank:

- Recorg
- Keyoff
- Imbed
- Replicate
- CIsize Data
- % Freespace CI
- % Freespace CA
- Shareoptions Xregion
- Shareoptions Xsystem

If you specify the following values:

Avgrec = K
Avg Value = 80
Primary = 500
Secondary = 100
Directory = 100

you request 40000 KB of primary space, 8000 KB of secondary space, and 100 directory blocks. The JCL used to accomplish the same task looks like:

```
//DD1  ...   SPACE=(80,(500,100,100)),AVGREC=K   ...
```

Retpd or Expdt

specifies the retention period or expiration date that is associated with the data class being defined. Retention period is the number of days (0 to 9999) and expiration date is the date when you want the definition to expire. The default is 'blank' for Expdt and 0 for Retpd, no expiration time.

Volume Count

specifies the maximum number of SMS-managed volumes that a data set can span, unless they are overridden by sustained data rate (SDR) or increased by external volume additions or internal additions, up to the dynamic volume count. (See the description of the Dynamic Volume Count attribute in "Defining Volume and Data Set Attributes for Data Class" for the cases in which that attribute is valid.)

Volume count is ignored for data sets to which no storage class is assigned. The default is 1.

Note: In JES3 systems, volume count is also ignored for managed mountable data sets.

Add'l Volume Amount

specifies whether primary or secondary allocation amounts are to be used when the data set is extending to a new volume. You can specify:

- P for primary
- S for secondary

If you leave the field blank, the system default of primary is used. This attribute is used during VSAM EOV processing, and is only applicable to any VSAM multivolume data sets allocated in the extended format.

Defining Volume and Data Set Attributes for Data Class

From Page 1 of the Data Class Define panel, issue the DOWN command to view Page 2, shown in Figure 38 on page 116.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              DATA CLASS DEFINE                   Page 2 of 4
Command ===>

SCDS Name . . . : Y421252.MYSCDS
Data Class Name : DCA11

To DEFINE Data Class, Specify:

   Data Set Name Type  . . . . . .      (EXT, HFS, LIB, PDS or blank)
     If Ext  . . . . . . . . . . .      (P=Preferred, R=Required or blank)
     Extended Addressability . . . N    (Y or N)
     Record Access Bias  . . . . .      (S=System, U=User or blank)
   Space Constraint Relief . . . . N    (Y or N)
     Reduce Space Up To (%)  . . .      (0 to 99 or blank)
     Dynamic Volume Count  . . . .      (1 to 59 or blank)
   Compaction  . . . . . . . . . .      (Y, N, T, G or blank)
   Spanned / Nonspanned  . . . . .      (S=Spanned, N=Nonspanned or blank)

  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 38. Defining Data Class Volume and Data Set Attributes

You can specify the following attributes on the second page of the Define Data Class panel:

Data Set Name Type

specifies the format in which the data set is to be allocated. When you specify extended format, you can also select requirements for the data set, including the need for extended addressability.

See *z/OS DFSMS: Using Data Sets* for detailed information on assigning and allocating data classes for extended format data sets.

The available values are:

EXT specifies that the data set to be allocated is in the extended format. All VSAM data set types can be allocated in the extended format (with the exception of key range data sets, temporary data sets, and system data sets, including all catalogs).

If you specify EXT, you must also specify a value for the If Ext subparameter. The other subparameters are primed with the following defaults: Extended Addressability = No, and Record Access Bias = USER.

Restriction: You cannot use JCL, ALLOCATE, or DEFINE to specify EXT.

HFS specifies that the data set is a hierarchical file system (HFS) data set.

LIB specifies that the data set is a PDSE.

PDS specifies that the data set is a partitioned data set.

blank leaves the Data Set Name Type attribute value unspecified.

If you specify a Data Set Name Type of EXT for data sets allocated in extended format, you can specify additional attributes:

If Ext specifies whether allocation in extended format is preferred or required. If you specified EXT for the Data Set Name Type attribute, you must also specify one of the following:

P (preferred) The data set allocation is attempted in nonextended format if the necessary system resources for extended are not available.

R (required) The data set allocation fails if the data set cannot be allocated in extended format.

Extended Addressability
specifies that a VSAM data set in the extended format be allocated using extended addressability. This lets the data set grow beyond the four gigabyte (4GB) size. The data set can be a VSAM data set in any record organization but must be allocated in the extended format.

The available values are:
Y Extended addressability is used.
N Extended addressability is not used. This is the default.

Record Access Bias
specifies whether to let VSAM determine how many and which type of buffers to use when accessing VSAM extended format data sets by batch processing. This is known as system-managed buffering, and is available to VSAM data sets in any record organization which are allocated in the extended format.

Tip: The buffering algorithms determined by VSAM can be overridden via JCL.

The available values are:

System specifies that VSAM chooses how many buffers to use and how they are processed.

System-managed buffering only takes effect if the application requests the use of non-shared resources (NSR). It is not effective for applications requesting local shared resources (LSR), global shared resources (GSR), or record-level sharing (RLS).

The number of buffers and the buffering technique are determined by the system or the user:

- The system, based on application specifications (ACB MACRF=(DIR,SEQ,SKP)) and the values for direct and sequential millisecond response (MSR) and bias specified in the storage class.
- The user, as specified on the JCL DD card (AMP=('ACCBIAS=).

If sequential processing is to be used, the system optimizes the number of buffers and uses the NSR buffering technique. If direct processing is to be used, the system optimizes the number of buffers and uses the LSR buffering technique. You can change the defaults for space and the relative amount of hiperspace when direct optimization is

used by using the following keywords in the AMP= parameter: SMBVSP=, SMBHWT=, and SMBDFR=.

User specifies that system-managed buffering is not used. This is the default.

Dynamic Volume Count
is used during allocation processing to determine the maximum number of volumes a data set can span.

During define processing, Dynamic Volume Count allows for a larger number of volumes to be considered without increasing the number of candidate volumes stored in the catalog. During existing data set allocation, it provides a way to increase the number of TIOT/JFCB entries that are created, so that more volumes can be dynamically allocated as required during the lifetime of the allocation.

For VSAM data sets, the Dynamic Volume count is the maximum number of volumes that all components in the sphere being allocated can span.

All the following must be true for the Dynamic Volume Count value to be valid:

- The Dynamic Volume Count is larger than the current volume count of the data set.
- The data set is SMS-managed.
- The Space Constraint Relief value is Y.

The Dynamic Volume Count field overrides any administrator-specified data class Volume Count, and any user-specified volume information provided via JCL, IDCAMS, or TSO. The default is 1.

Compaction
specifies whether data is to be compressed. You can compress data on tape, or on DASD if the data set is allocated in the extended format. This field is ignored for DASD data sets if the data set name type is not EXT. A compressed data set cannot reside on the same cartridge as a data set that is not compressed.

For physical sequential data sets, you can specify T or G to compress using either tailored or generic compression dictionaries. This overrides the compression option set in PARMLIB, and lets you select the type of compression on a data set level. Current users of generic dictionaries can move to using tailored dictionaries a data set at a time, as new data sets are created.

Spanned / Nonspanned
specifies whether a data record can span control interval boundaries. This applies to both system-managed and non-system-managed data sets. Specify one of the following:

Spanned specifies that if the size of a data record is larger than a control interval, the record can be contained on more than one control interval. This lets VSAM select a control interval size that is optimum for the DASD.

When a data record that is larger than a control interval is put into a cluster defined for spanned record format, the first part of the record fills a control interval. Subsequent control intervals are filled until the record is written into the

cluster. Unused space in the record's last control interval is not available to contain other data records.

Restriction: Do not use this attribute for a variable-length relative record data set (VRRDS).

Nonspanned specifies that a record must be contained in one control interval. VSAM selects a control interval size that accommodates the largest record in the data set. This is the default.

Specifying Attributes to Handle Space Constraints During Allocation

You can specify attributes on Page 2 of the Data Class Define panel to indicate whether or not to retry new data set allocations or extends on new volumes that fail due to space constraints.

During allocation, there might not be enough space on a volume to meet the requested space. SMS volume selection can sometimes solve this problem by trying all candidate volumes before failing the allocation. You can also use the Space Constraint Relief and Reduce Space Up To (%) attributes to request that an allocation be retried if it fails due to space constraints. SMS retries the allocation by combining any of the following:

- Spreading the requested quantity over multiple volumes
- Allocating a percentage of the requested quantity
- Using more than 5 extents

Space Constraint Relief
specifies whether or not to retry an allocation that was unsuccessful due to space constraints on the volume. Before it is retried, the allocation is attempted on all candidate volumes. Space Constraint Relief applies only to system-managed data sets, and is limited to new data set allocations, and while extending the data set on new volumes. Specify one of the following:

Y specifies that SMS retry the allocation.

N specifies that SMS does not retry the allocation, so that allocation is not attempted on multiple volumes.

This is the default.

If you specify Y, SMS begins the retry process. This is a one- or two-step process, depending on the volume count you specified. For JCL allocations, SMS determines the volume count by taking the maximum of the unit, volume, or volser count. If these are not specified, SMS picks up a volume count from the data class. If there is no data class, SMS defaults the volume count to 1.

- If the volume count is 1 (one-step process)

  SMS retries the allocation after reducing the requested space quantity based on the Reduce Space Up To attribute. SMS simultaneously removes the 5-extent limit, so that SMS can use as many extents as the data set type allows.
- If the volume count is greater than 1 (two-step process)

  First, SMS uses a *best-fit* volume selection method to spread the primary quantity over more than one volume (up to the volume count). If this fails, SMS continues with the best fit method after reducing the primary quantity and removing the 5-extent limit.

Tip: You can remove the 5-extent limit without reducing the primary quantity by specifying 0 for the Reduce Space Up To (%) attribute.

For extends to new volumes, space constraint relief is strictly a one-step process. If regular volume selection has failed to allocate space, SMS reduces space or removes the 5-extent limit, but does not try the best-fit method.

The number of extents vary depending on data set type, as follows:

- Non-VSAM, non-extended format data sets: up to 16 extents on the volume
- Non-VSAM, extended format data sets, up to 123 extents.
- PDSE and HFS, up to 123 extents on the volume
- VSAM data sets, up to 255 extents per component but only up to 123 extents per volume per component
- Striped VSAM data sets, up to 4080 extents per data component.

Reduce Space Up to (%)
specifies the amount by which you want to reduce the requested space quantity when the allocation is retried. You must specify Y for the Space Constraint Relief attribute. Valid values are 0 to 99.

If you request space constraint relief but do not specify a percentage value (either 0 or blank), SMS does not reduce the requested space quantity. This implies your application cannot tolerate a reduction in the space to be allocated, so only the 5 extent limit is relieved.

When you request space constraint relief in one or more data classes, you might notice any of the following:

- Very large allocations might succeed if a sufficiently large volume count is specified in the data class or through JCL.
- Existing data sets might end up with less space than initially requested on extents.
- The space allocated for new data sets might be less than requested.
- The number of extents used during initial allocation might result in fewer extents being subsequently available. For example, if the primary space allocation uses 10 extents when allocating a physical sequential data set, then only 6 extents are left for allocation of the secondary quantity.
- X37 abends should occur less frequently.

When you request space constraint relief in one or more data classes, you might notice any of the following:

- Very large allocations might succeed if a sufficiently large volume count is specified in the data class or through JCL.
- Existing data sets might end up with less space than initially requested on extents.
- The space allocated for new data sets might be less than requested.
- The number of extents used during initial allocation might result in fewer extents being subsequently available. For example, if the primary space allocation uses 10 extents when allocating a physical sequential data set, then only 6 extents are left for allocation of the secondary quantity.
- X37 abends should occur less frequently.

Defining VSAM Attributes and Specifying Media Types for Data Class

From Page 2 of the Data Class Define panel, use the DOWN command to view Page 3, shown in Figure 39:

```
  Panel  Utilities  Scroll  Help
 ------------------------------------------------------------------------------
                                 DATA CLASS DEFINE                 Page 3 of 4
 Command ===>

 SCDS Name . . . : Y421252.MYSCDS
 Data Class Name : DCA11

 To DEFINE Data Class, Specify:

   Media Interchange
     Media Type  . . . . . . . . . . .      (1, 2, 3, 4 or blank)
     Recording Technology  . . . . . . N    (18, 36, 128, 256 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  . . . . . . . . . .      (1 to 100 or blank)
               CA  . . . . . . . . . .      (1 to 100 or blank)

Use ENTER to perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit
```

Figure 39. Defining Data Class VSAM Attributes and Media Types

You can specify the following attributes on the third page of the Define Data Class panel:

Recorg specifies the data set organization, and it resembles the RECORG DD attribute. If you specify a Recorg value, you cannot specify the Recfm attribute. A blank value specifies either a physical sequential or a partitioned organization (non-VSAM data set).

If you do not specify a Recorg value for data sets with a data class, assigned either by JCL or ACS routine, the DSORG defaults to either physical sequential (PS) or partitioned organization (PO). Data class does not have a DSORG field. To specify a physical sequential data set, specify RECFM. To specify partitioned organization, specify Recfm and Space values for directory blocks. See *z/OS DFSMS: Using Data Sets*.

Table 7 summarizes which data class attributes apply to each record organization (RECORG).

Table 7. Applying Data Class Attributes to Record Organization (Recorg)

Attribute	Record Organization (RECORG)				
	Blank	**KS**	**ES**	**RR**	**LS**
LRECL	x	x	x	x	
RECFM	x				
KEYLEN	x	x			
KEYOFF	x	x			
VOLUME COUNT	x	x	x	x	x
SPACE	x	x	x	x	x
CISIZE	x	x	x	x	x
FREESPACE	x	x			
SHAREOPTIONS	x	x	x	x	x
Retpd or Expdt	x	x	x	x	x
DSNTYPE	x	x			
COMPACTION	x	x			

See the following for more information:
- *z/OS DFSMS Access Method Services*
- *z/OS DFSMS: Using Data Sets*
- *z/OS MVS JCL Reference*

Keylen specifies the key length in bytes. To use this attribute, you must specify either KS or blank for the Recorg attribute. If the Recorg value is KS, the Keylen attribute represents the length of the KSDS key field and ranges from 1-255, or you can leave it blank. If the Recorg value is blank, the Keylen value ranges from 0-255, or you can leave it blank. For either value of Recorg, you must assign a value to the Keylen attribute that is less than or equal to the LRECL value.

Keyoff specifies the displacement from the beginning of a record to the KSDS key field. It is valid only when the Recorg value is KS. The Keyoff value can range from 0 to the value of (LRECL - Keylen).

If a Keylen value is specified, then the Keyoff attribute must also be specified, or 0 value is used. This value is merged only if the Keylen value is not specified in JCL.

CIsize DATA specifies the control interval size for the data component of data sets having Recorg values of KS, ES, RR,or LS. The default is 4096.

% Freespace specifies the percentage of space to be left free when a cluster is loaded. It also specifies the percentage of space to be left free after any split of control intervals (CI) or control areas (CA). It applies only to the data component. The Freespace attribute is valid only if the Recorg value is KS or blank. The default is blank.

Media Interchange
provides the capability to control the type of media created to make the data and media acceptable to other processors in the same or different locations. For example, you can create a data class with a name of BOSTON and define the media interchange parameters (Media Type and Recording Technology attributes) compatible with the tape hardware in the Boston location. When the hardware in Boston is changed, the BOSTON data class can be changed to the new media interchange parameters.

Media Type specifies mountable tape media cartridge type. Valid values are MEDIA1, MEDIA2, MEDIA3, MEDIA4, or blank. This field is optional.

Recording Technology
specifies recording technology for mountable tape media cartridges. This field is optional and dependent on the value specified in the Media Type field:

- If MEDIA1 is specified, either 18TRACK or 36TRACK is valid.
- If MEDIA2 is specified, only 36TRACK is valid.
- If MEDIA3 or MEDIA4 is specified, either 128TRACK or 256TRACK is valid.

  The system uses 128TRACK by default if you specify MEDIA3 or MEDIA4 without defining a value in the Recording Technology field.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for further information on data class attributes and the Recording Technology field.

Block Size Limit
specifies the block size limit for new tape data sets on SMS and non-SMS managed storage. This value can be from 32760 - 2147483648, 32 KB - 2097152 KB, 1 MB - 2048 MB, or 1 GB - 2 GB. The default is blank.

When a program opens a tape data set without a block size value for output, the system determines a block size. This system-determined block size value is less than or equal to the first value in the following:

- BLKSZLIM keyword on the DD statement or the dynamic allocation equivalent
- Block size limit specified in data class (even for non-SMS-managed data sets)
- TAPEBLKSZLIM keyword in the DEVSUPxx member of SYS1.PARMLIB
- 32760

Currently, OPEN chooses a block size value of no greater than 262144, or 256 KB. Any larger limit has no effect.

Defining Shareoptions and RLS Attributes for Data Class

From Page 3 of the Data Class Define panel, use the DOWN command to view Page 4, shown in Figure 40 on page 124:

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

 SCDS Name . . . : Y421252.MYSCDS
 Data Class Name : DCA11

 To DEFINE Data Class, Specify:

   Shareoptions Xregion. . . . . . . .      (1 to 4 or blank)
                Xsystem  . . . . . . .      (3, 4 or blank)
   Reuse . . . . . . . . . . . . . . . N    (Y or N)
   Initial Load  . . . . . . . . . . . R    (S=Speed, R=Recovery or blank)
   BWO . . . . . . . . . . . . . . . .      (TC=TYPECICS, TI=TYPEIMS, NO or blank)
   Log . . . . . . . . . . . . . . . .      (N=NONE, U=UNDO, A=ALL or blank)
   Logstream Id  . . . . . . . . . . .      ____________________
   FRlog . . . . . . . . . . . . . . . N    (N=NONE, R=REDO or blank)
   RLS CF Cache Value  . . . . . . . . A    (A=ALL, N=NONE, U=UPDATESONLY)

Use ENTER to perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit
```

Figure 40. Defining Data Class Shareoptions and RLS Attributes

You can specify the following attributes on the fourth page of the Define Data Class panel:

Shareoptions
specifies how end users can share a component or cluster.

The Xregion value specifies the amount of sharing allowed among regions within the same system, or within multiple systems using global resource serialization (GRS). You can specify the following values:

1 Any number of users can read the data set at one time, *or* only one can write to it. This setting does not allow any type of non-RLS access when the data set is already open for RLS processing.

2 Any number of users can read the data set at one time, *and* only one can write to it. If the data set is already open for RLS, non-RLS users can read the data set but cannot write to it. If the data set has been opened for non-RLS output, an RLS open fails.

Requirement: You must apply APARs OW25251 and OW25252 to allow non-RLS read access to data sets already open for RLS processing.

3 Any number of users can share the data set, and each is responsible for maintaining read and write integrity. This setting does not allow any type of non-RLS access when the data set is already open for RLS processing.

4 Any number of users can share the data set, and buffers used for direct processing are refreshed for each request. This setting does not allow any type of non-RLS access when the data set is already open for RLS processing.

The Xsystem value specifies the amount of sharing allowed among systems. You can specify the following values:

3 Any number of users can share the data set, and each is responsible for maintaining read and write integrity.

4 Any number of users can share the data set, and buffers used for direct processing are refreshed for each request.

If values are not specified explicitly or in the data class, VSAM defaults are used.

RLS CF Cache Value
specifies the amount of data that will be cached. This keyword is honored only when RLS_MaxCfFeatureLevel(A) is active in the sysplex. For more information about the RLS_MaxCfFeatureLevel parameter, see 21.

DFSMS CF cache structures are connected to the system when the first VSAM RLS instance is opened on the system. At this time, message IGW500I is issued to indicate whether the RLS CF Cache Value keyword is honored.

You can specify the following values:

ALL indicates that VSAM index and data components will be cached. **ALL** is the default.

NONE indicates that only the VSAM index data will be cached. The data components will not be placed in the cache structure.

UPDATESONLY
indicates that only WRITE requests will be placed in the cache structure.

Specifying Attributes for Data Set Reuse and Loading

On Page 4 of the Data Class Define panel, you can also specify attributes to indicate whether the VSAM cluster can be reused and how the data set is to be loaded.

Reuse specifies whether the VSAM cluster can be opened again as a new data set. Specify one of the following:
Y (Yes) The cluster is reusable.
N (No) The cluster is not reusable.

Initial Load specifies how the data set is to be loaded. Specify one of the following:

S (Speed) The data set is loaded without being preformatted.

R (Recovery) The data set is preformatted when it is loaded. This is the default.

Specifying Attributes for Backup-While-Open (BWO) and Recovery

You can specify whether a data set is eligible for backup-while-open (BWO) processing. You can also indicate whether a data set is recoverable or not, and if so, provide the name of the forward recovery log stream.

Restriction: These attributes are only available to system-managed data sets.

BWO specifies whether BWO is to be used. This applies only to system-managed VSAM data sets, and is not available for linear data sets. Specify one of the following:
TYPECICS BWO processing is used for CICS™ VSAM file control data sets. If you specify BWO(TYPECICS), CICS checks to see that you have also specified

LOG(ALL) and that a logstream ID is available, either through the access method services DEFINE command or through the data class. SMS rejects dynamic or JCL allocations if, as the result of an alter function, the value for LOG is changed to ALL without a logstream ID being available, or if the logstream ID is nullified.

TYPEIMS BWO processing is used for IMS VSAM data sets.

Note: Support for this option is only available with IMS 6.1 or above.

NO BWO is not used for CICS VSAM file control or IMS VSAM data sets. This is the default.

FRlog specifies whether VSAM batch logging is to be performed for your VSAM data set. Specify one of the following:

NONE disables the VSAM batch logging function for your VSAM data set. Changes made by applications are not written to the MVS log stream indicated on the logstream ID parameter.

REDO enables the VSAM batch logging function for your VSAM data set. Changes made by applications are written to the MVS log stream indicated on the logstream ID parameter.

blank The FRlog value in the catalog is used.

If you specify FRlog(REDO), you must specify the logstream ID parameter for the VSAM data set(s). If you do not specify the logstream ID, message IEC161I is issued.

There is no default JCL value. If FRlog is omitted, the catalog value is used.

Log specifies whether the data set is considered recoverable or not.

For data sets defined using access method services, the Log and Logstream ID attributes in the data class are merged with those defined in the DEFINE command. If a single logstream is to be used for each VSAM data set, the logstream ID should be specified on the DEFINE command. Specify the logstream ID in the data class only if the same logstream ID is to be used for many data sets. Otherwise, this will result in too many data classes.

Additionally, SMS rejects dynamic or JCL allocations if, as the result of an alter function, the value for LOG is changed to ALL without a logstream ID being available, or if the logstream ID is nullified. Specify one of the following:

NONE indicates that neither an external backout nor a forward recovery capability is available, so the data set is not considered recoverable

UNDO indicates that changes can be backed out using an external log, so the data set is considered recoverable

ALL indicates that changes can be backed out and forward recovered using an external log

If you specify Log(ALL), you must specify a logstream ID, either on the access method services DEFINE command or in the Logstream ID field in the data class.

blank the data set is not recoverable. This is the default.

Logstream ID identifies the CICS forward recovery log stream. It applies to all components of the sphere. If you specify Log(ALL) or FRlog(REDO), you must specify a logstream ID.

A logstream ID is made up of 1-to-26 characters, including separators. This name is made up of one or more segments, each containing one to eight alphabetic, numeric, or national characters. The first character of each segment must be an alphabetic or national character. Segments are joined by periods.

For data sets defined using access method services, the attributes in the data class are merged with those defined in the DEFINE command. If a single logstream is to be used for each VSAM data set, the logstream ID should be specified on the DEFINE command. Specify the logstream ID in the data class only if the same logstream ID is to be used for many data sets. Otherwise, this will result in too many data classes.

Assigning Data Classes

You can define an ACS routine to determine data classes, or end users can explicitly specify a data class name on the following:

- JCL DD statements
- TSO/E ALLOCATE commands
- Access method services ALLOCATE and DEFINE commands
- Dynamic allocation requests such as through ISPF/PDF data set allocation panels

End users can explicitly specify data set attributes, and these specifications take precedence over the data set attributes assigned through the data class ACS routine.

Specifying Data Classes outside ACS Routines

The syntax for specifying a data class on a JCL statement is:

```
DATACLAS=data-class-name
```

The format for specifying a data class on a TSO/E command is:

```
DATACLAS(data-class-name)
```

The format for specifying a data class on an access method service command is:

```
DATACLAS(-)
```

Processing Data Class Attributes in JCL

The order of precedence for data class attributes in JCL is as follows:

1. Explicit specifications
2. LIKE and REFDD keywords
3. Data class definitions (explicit or derived) in REFDD statement
4. Data class definitions in the referencing DD

A data class does not need to be a self-contained, complete data organization. You can partially define the data set attributes in the data class definition (as a base) and the user can explicitly specify the remaining attributes. However, the merging of all attributes according to the order specified above must result in a valid data organization. In the example below,

```
//DD1  DD    ...,DATACLAS=DC2,RECORG=ES,...
//DD2  DD    ...,DATACLAS=DC1,LRECL=180,REFDD=DD1,...
```

For the attributes for DD2, LRECL of 180 is used first, and then the RECORG of ES in DD1 is used, regardless of the values specified in either the DC1 or the DC2 data class. For the remaining attributes that are not explicitly specified on the DD statements, SMS uses the values defined in the data class definition of DC2 and then DC1.

In this next example, the attributes of the data set referenced by the LIKE keyword are used after all other explicit specification but before data class DC3 attributes:

```
//DD3  DD    ...,DATACLAS=DC3,LRECL=180,LIKE=SAMPLE.DATA,....
```

Here, an LRECL of 180 is used. Then, SMS uses the Data Set Control Block (DSCB) information from the SAMPLE.DATA data set. Finally, the remaining attribute values are drawn from the DC3 data class.

A final example illustrates the use of REFDD:

```
//DD4  DD   ...,DATACLAS=DC4,...
//DD5  DD   ...,DSN=DS1,REFDD=DD4,DATACLAS=DC5,LRECL=180,...
```

In this example, the REFDD keyword specifies that the explicit attributes on the DD4 JCL statement are to be used second since explicit attributes on the DD4 JCL statement are used after the explicitly specified attributes. Next, the attributes from data class DC4 referenced in DD4 should be used. The remaining attributes are taken from data class DC5.

You can use the DATACLAS keyword to specify data class attributes for SMS-managed or non-SMS-managed data sets. Or, you can use the DCB=*.ddname or DCB=dsname keywords to copy the attributes of existing data sets for new non-SMS-managed data set allocations. You can specify the attributes of new SMS-managed data sets with the LIKE or REFDD keyword. The LIKE and REFDD keywords are mutually exclusive.

Restriction: You cannot use the REFDD keyword to copy DCB attributes from the DSCB. Use the LIKE keyword to copy from the DSCB.

Related Reading: For more information on using the LIKE and REFDD keywords, see *z/OS MVS JCL Reference*. For information on determining data classes through ACS routines, see “Chapter 9. Defining ACS Routines” on page 137. For information on access method services DEFINE defaults and data class defaults, see *z/OS DFSMS Access Method Services*.

Defining Additional Data Classes

You can copy existing data classes and modify them to create new ones by using the COPY line operator from the List panel, which is explained in “Copying SMS Classes, Storage Groups, and Aggregate Groups” on page 209.

Chapter 8. Defining Aggregate Groups

Individual SMS data sets in pool storage groups are backed up based on the backup attributes that are in a management class. Using ISMF's aggregate group application, you can group and back up data sets according to application or backup requirements. This chapter describes aggregate groups and shows you how to define them using the ISMF aggregate group application.

Understanding Aggregate Groups

An aggregate group is an SMS construct that uses control information and data set lists to define an application or other group. The data stored in the aggregate group is the principal input to DFSMShsm application backup and recovery.

An aggregate group consists of backup criteria and a group of data sets selected for backup by the storage administrator according to application or other requirements. See *z/OS DFSMShsm Data Recovery Scenarios* for further information on backing up or recovering aggregate groups.

The aggregate group application allows you to:

- Generate a list of aggregate groups
- Display the attributes of a single aggregate group
- Define or alter the attributes of a single aggregate group
- Delete aggregate groups
- Edit and browse the selection data sets associated with aggregate groups
- Edit and browse the instruction data sets associated with aggregate groups
- Back up the selected aggregate group
- Recover an aggregate group that has already been backed up
- Specify one to fifteen local copies of aggregate backup (ABACKUP) output files to be created
- Assign aggregate backup attributes to an aggregate group by specifying a management class name

Restriction: You cannot assign an object to an aggregate group.

ABACKUP allows specifying the number of copies of the Aggregate Backup and Recovery Support (ABARS) output files to be created using the ISMF panel shown in Figure 42 on page 132. The Copies value can be 1-15. The default is 1.

In order to support the creation of multiple copies, a unique name must be created for each copy. This entails using a new suffix convention for the output data sets:

- .D.CccVnnnn for the DFSMSdss data file
- .O.CccVnnnn for the internal data file
- .C.CccVnnnn for the control file
- .I.CccVnnnn for the instruction or activity log file

where

cc represents the copy number (a number from 1-15 of the copy created)

nnnn represents the version number

The name supports the elimination of the GDG support.

© Copyright IBM Corp. 1986, 2002

Planning Aggregate Groups

First, you should identify those applications that are vital to continuing operation. Then, you must identify the application's associated components. These include:

- JCL and procedures
- Source, object, and load module data sets
- Procedures and run books
- Required system data sets
- Application data

Next, you should identify the required primary and recovery locations to ensure that the operating environments are compatible and the necessary resources are available. A primary location is the location that does an aggregate group backup of a specific application. A recovery location is the location where the aggregate group recovery of that specific application is performed.

Last, you should consider data set naming conventions to avoid duplicate data set names, and if system-managed data sets are to be recovered, you must ensure that the SMS constructs and attributes are compatible. Once the data sets needed to recover an application are identified, you can define the aggregate group.

Defining Aggregate Groups

You can use ISMF to define, alter, list, display, back up or recover an aggregate group, or edit the selection or instruction data sets associated with an aggregate group by selecting option 9, Aggregate Group, from the ISMF Primary Option Menu for Storage Administrators. A selection data set is a data set which contains lists of data sets to be included in the backup of an application or other group. An instruction data set is a data set which contains instructions, commands, and so on, that are copied into the control file volume after the backup control file.

See the following for further information on the options:

- "Listing SMS Classes, Aggregate Groups, Storage Groups, and Libraries Using ISMF" on page 195 for additional information on option 1
- "Altering Aggregate Groups" on page 206 for additional information on option 4
- "Backing up and Recovering an Aggregate Group" on page 135 and *z/OS DFSMShsm Data Recovery Scenarios* for additional information on options 5 and 6.

Figure 41 on page 131 illustrates the Aggregate Group Application Selection panel.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                     AGGREGATE GROUP APPLICATION SELECTION
Command ===>


To Perform Aggregate Group Operations, Specify:
  CDS Name . . . . . . . . 'SMS.SCDS1.SCDS'
                                (1 to 44 Character Data Set Name or 'Active')
  Aggregate Group Name . . TESTAG    (for Aggregate Group List, fully or
                                     Partially Specified or * for All)
Select one of the following Options:
  3  1. List     - Generate a list of Aggregate Groups
     2. Display  - Display an Aggregate Group
     3. Define   - Define an Aggregate Group
     4. Alter    - Alter an Aggregate Group
     5. Abackup  - Backup an Aggregate Group
     6. Arecover - Recover an Aggregate Group

If List Option is Chosen,
  Enter "/" to select option      Respecify View Criteria
                                  Respecify Sort Criteria

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

Figure 41. Aggregate Group Application Selection

To define an aggregate group, you must specify a CDS Name and an Aggregate Group Name on the panel and select option 3, Define. The CDS Name must be the name of an SCDS. ISMF primes the CDS Name field with the name last used for an aggregate group. The default is 'Active,' which represents the currently active configuration, but you cannot define or alter aggregate groups to the 'Active' configuration.

In Aggregate Group Name, you must specify the name of the aggregate group that you are defining. ISMF primes the field with the name last used.

Defining Aggregate Group Attributes

Select option 3, Define, and press Enter to see the first of the two pages of the Aggregate Group Define panel shown in Figure 42 on page 132. You can leave either page of the Aggregate Group Define panel at any time without saving the aggregate group attributes or changes by issuing the Cancel command.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                            AGGREGATE GROUP DEFINE              Page 1 of 2
Command ===>

SCDS Name  . . . . . : SMS.SCDS1.SCDS
Aggregate Group Name : TESTAG

To DEFINE Aggregate Group, Specify:
 Description ==>
             ==>

 Backup Attributes
   Number of Copies  . . . . . . . 1                (1 to 15)
   Management Class Name . . . . .                  (1 to 8 characters, to be
                                                    defined in current SCDS)

 Output Data Set Name Prefix . . .
                                                    (1 to 33 Characters)
 Account . . . . . . . . . . . . .
                                                    (1 to 32 Characters)

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

Figure 42. Aggregate Group Define, Page 1 of 2

Page one of the Aggregate Group Define panel contains aggregate group define attributes. The SCDS Name and Aggregate Group Name fields are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the aggregate group.

You can specify the following *required* attributes on the first page of the Aggregate Group Define panel:

Number of Copies
: specifies the number of aggregate backup output files to be created. The valid values are 1-15.

Management Class Name
: specifies the management class name from which the Aggregate Backup attributes are obtained. The valid values are 1-8 alphanumeric characters (first character not a digit) or a blank.

Output Data Set Name Prefix
: identifies the output data sets created by aggregate backup.

Editing Aggregate Group Attributes

After specifying your backup attribute values, issue the DOWN command to view the page two of the Aggregate Group Define panel, which is shown in Figure 43 on page 133.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                            AGGREGATE GROUP DEFINE              Page 2 of 2
Command ===>


SCDS Name  . . . . . : SMS.SCDS1.SCDS
Aggregate Group Name : TESTAG
To Edit a Data Set, Specify Number  . .     (1, 2, 3, 4, 5, or 6)
  Selection Data Sets:            (1 to 44 characters)
   1 ===>
     Member Name ===>             (1 to 8 characters)
   2 ===>
     Member Name ===>             (1 to 8 characters)
   3 ===>
     Member Name ===>             (1 to 8 characters)
   4 ===>
     Member Name ===>             (1 to 8 characters)
   5 ===>
     Member Name ===>             (1 to 8 characters)

  Instruction Data Set:           (1 to 44 characters)
   6 ===>
Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 43. Aggregate Group Define, Page 2 of 2

Page two of the Aggregate Group Define panel contains the selection and instruction data set names for the aggregate group. the SCDS Name and Aggregate Group Name fields are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel.

Edit a Data Set

Select the number of a selection or instruction data set that you want to edit. When you select a data set number, it allows you to allocate or modify the data set by invoking PDF Edit. The PDF edit screen is shown in Figure 44 on page 134. See *z/OS ISPF User's Guide Volume I* for more information on the PDF Edit commands.

Selection Data Set Name

Name of the data set containing lists of data sets to be included in the application backup. You can specify up to five selection data set names. One data set name is required. There is no default. If you want to enter a fully qualified data set name, enclose the name in single quotes. If you do not enclose the name in single quotes, the TSO prefix is added to the name as the first high level qualifier. There is no default.

Member Name

Name of the data set member containing lists of data sets to be included in the application backup if the selection data set is partitioned. This name must be a valid TSO data set member name. Enter a valid member of the partitioned data set specified in the Selection Data Set Name field. This is required if the data set specified in the Selection Data Set Name field is a partitioned data set. If you want to enter a fully qualified data set name, enclose the name in single quotes. If you do not enclose the name in single quotes, the TSO prefix is added to the name as the first high level qualifier. There is no default.

Instruction Data Set Name

Name of the data set containing instruction, commands, etc., that are copied into the control file volume after the backup control file. This data set can only be a sequential data set. You must use a valid TSO data set name. The data set name, including the TSO prefix, can be no more than 44 characters long. This is an optional field and has no default.

If you select the option to edit a Selection or Instruction Data Set on the Aggregate Group Define panel, you get the PDF Edit screen shown in Figure 44.

```
EDIT --- SELECT.DATASET.ONE ------------------------- LINE 000000 COL 001 072
COMMAND ===>                                              SCROLL ===> HALF
******  **************************** TOP OF DATA ******************************
000001 INCLUDE( XMP.** )
000002 EXCLUDE( XMP.USER.TAPE1,
000003          XMP.INPUT.MASTER )
000004 ACCOMPANY( DATA.MASTER,          /* MASTER */
000005            XMP.USER.TAPE1 )
000006 ALLOCATE( XMP.OLD.DASD1 )
****** *************************** BOTTOM OF DATA ******************************
```

Figure 44. Editing an Aggregate Group Selection Data Set- Example

You can specify the following keywords with parameters using PDF edit:

INCLUDE

specifies which data sets are included in the backup. No distinction is made whether they are tape or DASD. If a partitioned data set is indicated, all members are included in the application backup.

EXCLUDE

specifies the data sets that are specifically excluded from the backup process.

ACCOMPANY

specifies data sets that are physically removed from the backup site, transported to the recovery site, and only need to be cataloged during application recovery.

ALLOCATE

allocates and defines data sets at the recovery site without copying the data from the source. For VSAM data sets, only the base cluster is defined, and the AIXs and pathnames are not. See *z/OS DFSMShsm Storage Administration Guide* for further information.

When you allocate or edit selection data sets, there are several rules you must follow:

- Records must be 80 bytes in length and of a fixed format.
- Entries must be entered between columns 1 and 72.
- INCLUDE, EXCLUDE, ACCOMPANY, and ALLOCATE can be specified only once.
- A comment is a string of characters preceded by "/*" and followed by "*/" and might span multiple records.
- A separator consists of a comma (,), one or more blanks, or a comment.
- Parameters are separated from one another by one or more separators.

- One or more blanks can, optionally, precede and follow each parenthesis in the pair.
- Continuation of a record is optionally specified by a hyphen (-) or plus sign (+) as the rightmost nonblank character, preceded by one or more blanks. Continuation characters are not required.
- If the same fully qualified data set name is specified in both the INCLUDE list and the EXCLUDE list, the data set is not selected for processing because EXCLUDE takes precedence over INCLUDE.
- Data set names specified in the INCLUDE list cannot be specified in the ALLOCATE or ACCOMPANY list.
- The minimum truncation allowed for keywords is: **I** for INCLUDE, **E** for EXCLUDE, **AC** for ACCOMPANY, and **AL** for LOCATE.

Backing up and Recovering an Aggregate Group

To back up an aggregate group, select option 5, ABACKUP, on the Aggregate Group Application Selection panel, or enter the ABACKUP line operator on the Aggregate Group List panel, and press Enter. You get the Aggregate Group Backup panel.

To recover an aggregate group, select option 6, ARECOVER, and press Enter. You get the Aggregate Group Recover panel.

DFSMShsm handles your request for backup or recovery. You are prompted to enter information for backing up or recovering aggregate groups.

Defining Additional Aggregate Groups

You can copy existing aggregate groups and modify them to create new aggregate groups by using the COPY line operator, which is explained in “Copying SMS Classes, Storage Groups, and Aggregate Groups” on page 209.

Chapter 9. Defining ACS Routines

This chapter helps you define ACS routines. Specifically, it explains how you can write ACS routines for an SMS configuration using the ISMF Automatic Class Selection application. In doing so, it also provides general-use programming interface and associated guidance information.

Understanding ACS Routines

ACS routines can be used to determine the SMS classes and storage groups for data sets and objects in an SMS complex. For storage administrators, ACS routines automate and centralize the process of determining SMS classes and storage groups. ACS routines also help convert data sets to an SMS environment.

An object is assigned to a storage group when it is stored and remains in that storage group throughout its lifetime. The initial storage class and management class might be determined by defaults defined by an ACS routine or by explicit request. Storage class and management class assignments might be changed by the OSREQ CHANGE function or by automatic class transition. The OSREQ CHANGE request causes invocation of ACS routines that might override the requested assignments. During the automatic class transition, ACS routines are invoked to determine the new storage class and management class assignments.

“Chapter 15. Writing ACS Routines” on page 265 lists the rules for programming in the ACS language.

Tip: You can use the DFSMS NaviQuest tool to help you design and test your ACS routines. First, you can create test cases to perform extensive testing against test data representing actual data sets. Then you can test ACS routines in batch, freeing the workstation for other work. See “Appendix E. Using NaviQuest” on page 321for further information.

Through ISMF, you can create and maintain as many as four ACS routines in an SCDS, one for each type of SMS class and one for storage groups. After you have activated an SMS configuration, SMS executes ACS routines for the following operations:

- JCL DD statements (DISP=NEW, DISP=MOD)
- Dynamic allocation requests (DISP=NEW, DISP=MOD for a nonexistent data set)
- DFSMSdss COPY, RESTORE, and CONVERTV commands
- DFSMShsm RECALL and RECOVER
- Access method services ALLOCATE, DEFINE, and IMPORT commands
- OAM processing for STORE, CHANGE, and class transition
- Local data set creation by remote application through Distributed FileManager/MVS
- MVS data sets or z/OS UNIX System Service (z/OS UNIX) files created by remote application through the z/OS Network File System server

As a storage administrator, you write ACS routines using the ACS programming language, a high-level programming language. The language follows a logical, procedural flow of implementation that consists mainly of filtering criteria, IF/THEN statements, and SELECT/WHEN statements. Using these relational statements, ACS routines determine SMS classes and storage groups according to allocation parameters, data set sizes, object or data set names, and other variables.

© Copyright IBM Corp. 1986, 2002

All allocations directed to units that are neither tape nor DASD should be excluded from SMS management. Do this by testing for UNIT in the storage class routine and ensuring that the storage class is set to NULL in these cases.

Ensuring that no storage class is assigned for such allocations avoids potential errors with allocations that require specific types of units. For example, assigning a storage class to a VTAM® channel-to-channel (CTC) adaptor allocation results in sense errors when VTAM attempts to use the CTC.

Requirements: For system-managed data sets, the storage group is required because there is no way to explicitly specify storage groups. The other routines are optional for system-managed data sets. For objects, the storage group, storage class and management class ACS routines are required. For tape, the storage group, storage class and data class ACS routines are required.

Using ACS Routines for Data Sets Created by z/OS Network File System and Distributed FileManager/MVS

ACS routines can be used to determine the SMS classes for MVS data sets and z/OS UNIX files created by z/OS Network File System. When data sets and files are created by z/OS Network File System, any attributes related to SMS classes not specified by remote client user are defaulted using the ACS routines for the data set. If the remote user does not specify a storage class and if the ACS routines decide that the data set should not be SMS-managed, Distributed FileManager/MVS ends and returns an error to the remote workstation. Therefore, it is important that on the MVS target system, you consider the potential data set creation requests of remote users when constructing the ACS routines.

When data sets are created by Distributed FileManager/MVS, any attributes not specified by the source requester are defaulted using the attributes specified in the data and management classes selected by the ACS routines for the data set. If the system requires attributes which neither the remote user nor the data class specified, the creation process fails.

See “Determining Distributed FileManager/MVS Data Set Creation Requests” on page 282 for information on using ACS routines to determine the SMS classes for data sets created by Distributed FileManager/MVS.

Restrictions on Using ACS Routines

ACS routines are not invoked for of data sets that cannot be system-managed. These include the following:

- Dummy data sets
- Data sets that have SUBSYS or PATH coded on the JCL DD statement
- SYSIN and SYSOUT data sets

Creating ACS Routines

You must allocate either a sequential data set, a member of a partitioned data set (PDS), or a member of a partitioned data set extended (PDSE) for each of the ACS routines that you intend to write. As a general guideline, use an LRECL of 80. The maximum LRECL you can use is the maximum that can be specified on JCL. After allocating the data sets, select option 7, Automatic Class Selection Application, from the ISMF Primary Option Menu and press Enter.

ISMF displays the Class Selection Application Selection panel shown in Figure 45.

```
  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  . . 'SMS.SCDS1.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 45. Writing an ACS Routine

Select option 1, Edit, and press Enter to write an ACS routine.

ISPF/PDF displays the Edit Entry panel shown in Figure 46.

```
  Menu  RefList  RefMode  Utilities  LMF  Workstation  Help
-------------------------------------------------------------------------------
                              Edit Entry Panel
Command ===>


ISPF Library:
   Project . . .
   Group . . . .          . . .          . . .          . . .
   Type  . . . .
   Member  . . .                (Blank or pattern for member selection list)

Other Partitioned or Sequential Data Set:
   Data Set Name . . .
   Volume Serial . . .          (If not cataloged)

Workstation File:
   File Name . . . . .

Initial Macro  . . . .                 Enter "/" to select option
Profile Name . . . . .                 /  Confirm Cancel/Move/Replace
Format Name  . . . . .                    Mixed Mode
Data Set Password  . .                    Edit on Workstation
```

Figure 46. Invoking the ISPF/PDF Editor

After specifying your values, press Enter to invoke the ISPF/PDF Editor. Then enter the source code for the ACS routine. After entering the routine, issue the END command to save it and return to the ISPF/PDF Edit Entry panel. Use END again to return to the Automatic Class Selection Application Selection panel.

Translating ACS Routines

After creating an ACS routine, you must translate it into executable form. The translation process checks your source code for syntactic and semantic errors, generates an object form if no errors exist, and places the object form into the SCDS you specified on the translate panel. If the ACS routine that you are translating already exists in the SCDS, the new object form replaces the existing object form.

To translate an ACS routine, select option 2, Translate, from the Automatic Class Selection Application Selection panel. ISMF displays the Translate ACS Routines panel shown in Figure 47.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                              TRANSLATE ACS ROUTINES
Command ===>


To Perform ACS Translation, Specify:

   SCDS Name . . . . . . . 'SMS.SCDS1.SCDS'
                                    (1 to 44 Character Data Set Name)

   ACS Source Data Set . . 'SCDS1.STORGRP'
                                    (1 to 44 Character Data Set Name)

   ACS Source Member . . .          (1 to 8 characters)


   Listing Data Set  . . . 'STORGRP.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 47. Translating an ACS Routine

A successful translation places the ACS routine object table into the SCDS that you specify in SCDS Name. SCDS Name is a required field that is primed with the last SCDS that you have referenced or translated into.

Specify the name of the data set containing the ACS routine that you want to translate in ACS Source Data Set. This is a required field that is primed with the last used value.

ACS Source Member is the name of the member of the source data set that contains the ACS routine. The name is required only if the ACS source data set is a partitioned data set or PDSE. The field is primed with the last used value. The default is blanks.

Specify the name of a sequential data set to contain the translation results in Listing Data Set. If you specify a data set that already exists, the translation process replaces the existing data set contents with the results of the translation. If you specify a nonexistent data set, the translation process allocates space for it. If you leave this field blank, which is the default, you receive the results of the validation but you do not get a listing.

After specifying your values, press Enter to perform the translation.

Browsing the Results of a Translation

If you specify an ACS Source Data Set that contains no errors, you get a PDF Browse panel similar to the one shown in Figure 48 after translation.

```
 BROWSE -- STORGRP.LISTING--------------------------- LINE 00000000 COL 001 080
 COMMAND ===>                                            SCROLL ===> PAGE
******************************** TOP OF DATA  ********************************
ACS TRANSLATOR ***** TIME 09:45:32 DATE 11/19/1987 PAGE 0001 *****

SCDS NAME:              'SMS.SCDS1.SCDS'

ACS SOURCE DATA SET:     SCDS1.STORGRP
ACS SOURCE MEMBER:

0001      PROC STORGRP
0002        SELECT
0003          WHEN(&STORCLAS = DB*)
0004            SET &STORGRP = 'DATABASE'
0005          WHEN(&MAXSIZE > 9999999KB)
0006            SET &STORGRP = 'LARGE'
0007          WHEN(&DSTYPE = 'TEMP')
0008            SET &STORGRP = 'VIO','PRIMARY'
0009          OTHERWISE
0010            SET &STORGRP = 'PRIMARY'
0011        END
0012      END

TRANSLATION RETURN CODE:  0000
****************************** BOTTOM OF DATA ******************************
```

Figure 48. Browsing the Results of an ACS Translation

The Browse panel displays the SCDS name, the ACS routine source data set name, an output listing of the ACS routine, and a translation return code. SCDS Name, ACS Source Data Set, and ACS Source Member reflect the values that you specified on the Translate ACS Routines panel. The output listing contains the source code of the ACS routine and diagnostic messages. See *MVS/ESA System Messages, Volumes 1–5* for an explanation of the diagnostic messages.

The Translation Return Code displays one of the following codes:

0000 Successful translation.

0012 Unsuccessful translation. The ACS routine contains one or more semantic or syntactic errors. No object form was created and no updates were made to the SCDS.

0020 Internal error in translator.

When you exit from this panel, you get the Output Listing Disposition panel shown in Figure 49 on page 142.

```
  Panel  Utilities  Help
--------------------------------------------------------------------------------
                           OUTPUT LISTING DISPOSITION
Command ===>



Listing Data Set : STORGRP.LISTING


Specify Output Listing Disposition:

  Enter "/" to select option   _  Print Output Listing
                               _  Delete Output Listing









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

Figure 49. Generating an ACS Translation Listing

The short message area displays the results of the translation. Specify Y in Print Output Listing to submit a batch job to print the data set. Specify Y in Delete Output Listing to delete the data set after it is printed. Both fields are required, and they are primed with N. Table 8 summarizes the possible outcomes for each combination of Output Listing Disposition values.

Table 8. Results of Output Listing Disposition

Print	Delete	Resulting Action	Type of Job
Yes	Yes	Print data set `DISP=(OLD,DELETE)`	Batch
Yes	No	Print data set `DISP=(OLD,KEEP)`	Batch
No	Yes	Delete data set	Foreground
No	No	No action (dataset kept)	N/A

Issue the END command to return to the Translate ACS Routines panel.

BROWSING the Results of an Unsuccessful Translation

If you specified an ACS source data set that contains errors, and you specified a data set in the Listing Data Set field, the Browse panel shown in Figure 50 on page 143 is displayed after translation.

```
 BROWSE -- STORGRP.LISTING--------------------------- LINE 00000000 COL 001 080
 COMMAND ===>                                            SCROLL ===> PAGE
******************************** TOP OF DATA  ********************************
ACS TRANSLATOR ***** TIME 11:47:09 DATE 07/15/1991 PAGE 0001 *****

SCDS NAME:               'SMS.TEST.SCDS'
ACS SOURCE DATA SET:     SMS.ACS.ROUTINES
ACS SOURCE MEMBER:       SG1

0001      PROC STORGRP
0002        SELECT
0003          WHEN (&STORCLAS = DB*)
0004            SET &STORGRP = 'DATABASE'
0005          WHEN (&MAXSIZE > 99999999KB)
0006            SET &STORGRP = 'LARGE'
0007          WHEN &DSTYPE = 'TEMP'
*******   IGD03114I A LEFT PARENTHESIS WAS EXPECTED BUT NOT FOUND
0008            SET &STORGRP = 'VIO','PRIMARY'
0009          OTHERWISE
0010            SET &STORGRP = 'PRIMARY'
0011        END
0012      END

SUMMARY OF ERROR MESSAGES:

0007      IGD03114I A LEFT PARENTHESIS WAS EXPECTED BUT NOT FOUND

TRANSLATION RETURN CODE:  0012
*************************** BOTTOM OF DATA ***************************
```

Figure 50. Browsing the Results of an Unsuccessful ACS Translation

Scroll down to see a display of all the error messages together.

Validating ACS Routines or an Entire SCDS

SMS validates the entire SCDS when it is saved. You can validate the individual ACS routines of an SMS configuration after successfully translating them. You should also validate the entire configuration yourself so that you can see any error messages that result. Separate validation of the ACS routines does not produce all of the possible messages.

Validating an ACS Routine

To validate an ACS routine, select option 3, Validate on the Automatic Class Selection Application Selection panel. ISMF displays the Validate ACS Routines or Entire SCDS panel shown in Figure 51 on page 144.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                      VALIDATE ACS ROUTINES OR ENTIRE SCDS
Command ===>


To Perform Validation, Specify:

  SCDS Name  . . . . . 'SMS.SCDS1.SCDS'
                           (1 to 44 Character Data Set Name)

  ACS Routine Type . . SG  (DC=Data Class, MC=Management Class, SC=Storage
                           Class, SG=Storage Group, *=Entire SCDS)

  Listing Data Set . . SGVAL.LISTING
                           (1 to 44 Character Data Set Name)









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

Figure 51. Validating an ACS Routine

SCDS Name identifies the SCDS containing the ACS routine you are validating. It is a required field that is primed with the last used value. The default is blank.

ACS Routine Type identifies which of the four ACS routine types you are validating. When validating an individual ACS routine, the valid values are: DC, SC, MC, and SG. It is a required field that is primed with the last used value. The default is an asterisk, *, which specifies that you want to validate the entire SMS configuration in the SCDS, and not just one of the ACS routines.

Listing Data Set identifies the name of a sequential data set to contain the results of the validation. If you specify a data set that already exists, the validation process replaces the existing data set contents with the results of the validation. If you specify a nonexistent data set, the validation process allocates space for it. If you leave this field blank, which is the default, you receive the result of the validation, but you do not get a listing.

After specifying your values, press Enter to perform the validation. Validation fails if the following conditions are not satisfied (except where it is noted that only a warning is issued):

- **For the storage group ACS routine**:

  All of the defined pool storage groups should be possible outcomes of the routine.

  No dummy storage group can result from the storage group ACS routine.

  All storage groups that are possible outcomes of the storage group ACS routine must exist (be defined).

  All VIO storage groups are possible results of the storage group ACS routine.

  Notes:

  1. If you have no pool storage groups, you get a warning.
  2. If you have any pool type storage groups that are not possible outcomes of the storage groups ACS routine (that is, are not used in any SET statement),

you get a warning. This is because the storage group cannot be explicitly requested and therefore represents a wasted resource.

- **For the management class ACS routine**:

  All management classes that are possible outcomes of the management class ACS routine must exist.
- **For the storage class ACS routine**:

  All storage classes that are possible outcomes of the storage class ACS routine must exist.
- **For the data class ACS routine**:

  All data classes that are possible outcomes of the data class ACS routine must exist.

If an ACS routine uses the SET statement to assign an SMS class or storage group that does not exist, validation fails. If an ACS routine references an SMS storage class, management class, or data class that does not exist, SMS issues a warning. For example, you might delete a storage class. If you continue to reference the deleted storage class in the storage class ACS routine of an SCDS, SMS issues a warning when you validate the SCDS. If only warnings exist, the SCDS becomes valid despite the warning. This allows you to check for the deleted storage class and replace it with an existing storage class within the storage class ACS routine. For example:

```
WHEN (&STORCLAS = 'OLDSC')
  SET &STORCLAS = 'NEWSC'
```

If you specified a data set name in the Listing Data Set field, you get the Browse panel shown in Figure 52 after translation.

```
 BROWSE -- SGVAL.LISTING ---------------------------- LINE 00000000 COL 001 080
 COMMAND ===>                                                  SCROLL ===> PAGE
******************************** TOP OF DATA  *********************************
                            VALIDATION RESULTS

VALIDATION RESULT:  ERRORS DETECTED

SCDS NAME:          'SMS.SCDS1.SCDS'

ACS ROUTINE TYPE:   SG
DATE OF VALIDATION: 1987/11/19
TIME OF VALIDATION: 10:01

IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE DATABASE
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE LARGE
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE VIO
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE PRIMARY
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE PRIMARY
****************************** BOTTOM OF DATA ******************************
```

Figure 52. Browsing the Results of an ACS Routine Validation

VALIDATION RESULT can be VALIDATION SUCCESSFUL, ERRORS DETECTED or WARNINGS DETECTED. The SCDS Name and ACS Routine Type fields reflect the values you specified on the Validate ACS Routines or Entire SCDS panel. Browse also provides the date and time of validation. See *MVS/ESA System Messages, Volumes 1–5* for an explanation of the diagnostic messages.

When you leave Browse, you receive the Output Listing Disposition panel. See Figure 49 on page 142 and Table 8 on page 142 for information about printing and deleting.

Validating an Entire SCDS

For its contents to become the active storage management policy for an installation, an SCDS must be valid. Activating an SCDS validates its contents and copies them into the ACDS identified by IGDSMS*xx*. If the SCDS is not valid, activation fails.

You can validate an entire SCDS using the ISMF VALIDATE command. To use the command, type `VALIDATE` on the command line of the CDS Application Selection panel (shown in Figure 5 on page 28) and press ENTER. You receive the Validate ACS Routines or Entire SCDS panel shown in Figure 53.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                      VALIDATE ACS ROUTINES OR ENTIRE SCDS
Command ===>


To Perform Validation, Specify:

  SCDS Name  . . . . . 'SMS.SCDS1.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 . .
                           (1 to 44 Character Data Set Name)









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

Figure 53. Validating an SCDS

The SCDS Name field identifies the data set you want to validate. It is a required field that is primed with the last SCDS you have referenced or validated. You should place an asterisk (*) in the ACS Routine Type field, since you are validating an entire SCDS. You can save a listing of the validation results by specifying a sequential data set name in the Listing Data Set field. If you specify a data set that already exists, the old copy is replaced with your new listing. If you specify a nonexistent data set, ISMF allocates space for it. If you leave this field blank, you do not generate a listing.

To define a valid minimal SMS configuration, the configuration must contain:

- A fully defined base configuration
- A storage class definition
- A pool storage group containing at least one volume, or a VIO, object, or tape storage group
- A storage class ACS routine
- A storage group ACS routine

See *z/OS DFSMS: Implementing System-Managed Storage* for more information on defining a minimal SMS configuration.

Validation fails if any of the following conditions are not satisfied (except where it is noted that only a warning is issued).

- Base configuration information must exist. If you specify a default management class, you need to have defined the management class.
- Each possible outcome of the data class, storage class, management class, and storage group ACS routines must have a corresponding definition in the SCDS. For example, if you have five unique storage classes that can be determined by the storage class ACS routine, then you must have five corresponding storage class definitions in the SCDS.
- All classes and storage groups referenced in all ACS routines exist (issues a warning only).
- The storage group ACS routine exists.
- At least one VIO, pool, object type storage or tape group exists.
- All pool and dummy storage groups have at least one volume.
- At least one storage class exists.
- All pool and VIO storage groups are possible outcomes of the storage group ACS routine (issues a warning only).
- No dummy storage group is set by storage group ACS.
- All libraries associated with storage groups must exist in the configuration.
- Every optical library must have at least one drive associated with it.
- The DATABASE 2™ (DB2™) qualifier specified for each object or object backup storage group must be unique.
- Object and object backup storage groups can have up to eight real or pseudo optical libraries, but not both.
- An object or object backup storage group cannot reference a tape library.
- A tape storage group cannot reference an optical library.
- A tape storage group must have at least one tape library.
- If a default entry data class is specified, the data class must exist in the configuration.
- All cache sets specified in storage classes must be defined in the base configuration.

Translating and Validating in a Sysplex Environment

When running SMS in a sysplex with mixed levels of DFSMS, you must do the following on the system running the highest level of DFSMS:

- Define new SMS constructs and ACS routines
- Modify existing SMS constructs and ACS routines
- Translate your ACS routines
- Validate your SCDS

See “Altering the SCDS on Different DFSMS Releases” on page 209 for further information on altering the SCDS.

You can then activate and share the validated SCDS among systems with mixed levels of DFSMS. If you do not translate your ACS routines and validate your SCDS on the highest level of DFSMS, the translation and validation might fail. This failure occurs because some read-only variables are only known to higher levels of DFSMS. It can also occur due to changes in validation rules between releases.

Guideline: Coexistence PTFs allow you to share CDSs among mixed DFSMS levels and switch back to a prior release. However, once the CDSs have been formatted by the higher-level system, the CDS control blocks reflect the new rather than the down-level lengths.

When sharing the same SMS CDSs among mixed releases, or when switching back to a lower-level release using the same SMS CDSs which were formatted by the higher-level system, all coexistence PTFs *must* be applied to avoid corrupting the CDSs.

Testing ACS Routines

After completing your ACS routines, you can write and execute test cases using the ISMF Automatic Class Selection application. After testing the individual routines of a new or modified configuration, you can activate it with greater confidence. ACS installation exits are not invoked during ACS routine testing.

Restriction: During ACS testing, only those parameters that are passed to the ACS routines at ACS time are tested. For some data processing, what is specified in the JCL is not necessarily what is passed to the ACS routines. A good example of this is how the RETPD and EXPDT parameters work: if EXPDT is specified in the JCL, during normal processing the RETPD read-only variable is set based on the EXPDT. If your ACS routine logic bases decisions on RETPD variable, and you specify EXPDT as a test parameter, you might get unexpected test results. ACS testing is not designed to simulate all of the possible processing that might occur before ACS processing. You must take care to know what is passed to the ACS routines at ACS time to ensure that you are testing what you expect to occur on your system.

ACS Test Usage

Since a DSNTYPE of EXT cannot be specified in JCL, ACS test for an data set allocated in extended format has some considerations.

To test data class only, you should not specify DSNTYPE = EXT for extended format data sets since the value of DSNTYPE in the Data Class ACS routine is never set to EXT during actual ACS processing.

To test more than just the Data Class for extended format data sets, you can choose to test the Data Class ACS routine with a separate test case that has no DSNTYPE specified. To test other constructs, such as management class, storage class and storage group, you should use a test case with DSNTYPE set to EXT.

Note: ACS read-only variables are not defaulted from Data Class when the ACS test case is run (unlike actual ACS processing).

Creating ACS Test Cases

If you wish to test ACS routines, you need to allocate a partitioned data set for the ACS test cases. The partitioned data set must have a logical record length of 80 or greater and a record format of F or FB. After allocating the data set, go to the ISMF Primary Option Menu and select option 7, Automatic Class Selection Application. To specify test cases, select option 4, Test, from the Automatic Class Selection Application Selection panel. When you press Enter, you get the ACS Test Selection panel shown in Figure 54 on page 149.

```
  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 DEFIN E or ALTER Option is Chosen, Specify:

  ACS Test Library . . SCDS1.ACSTEST.LIB
  ACS Test Member  . . TEST1 ‍‍‍
```

Figure 56 shows page 2 of the ACS Test Case Define panel:

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                            ACS TEST CASE DEFINE                Page 2 of 4
Command ===>


ACS Test Library  : SCDS1.ACSTEST.LIB
ACS Test Member . : TEST1

To DEFINE ACS Test Case, Specify:
  Job  . . . . .          Pgm . . . . .            Vol . . . . .
  Group  . . . .          Applic  . . .            Unit  . . . .
  User . . . . .          Nvol  . . . .            Msvgp . . . .
  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 56. Creating ACS Test Cases, Page 2 of 4

Figure 57 shows page 3 of the ACS Test Case Define Panel:

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                            ACS TEST CASE DEFINE                Page 3 of 4
Command ===>

ACS Test Library  : USER2.TESTLIB
ACS Test Member . : TEST11

To DEFINE ACS Test Case, Specify:

  Dest . . . . . ____________________________________________
  Pool . . . . . ________
  Policy . . . . ________
  Parm     ====> ______________________________________________________
           ====> ______________________________________________________
           ====> ____________________________________________

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 57. Creating ACS Test Cases, Page 3 of 4

On this panel, you can specify values corresponding to the MSxxxx parameters which you expect to be returned from your tape management system through the

pre-ACS routine. See “Tape Management System Support” on page 163 for a description of these parameters, and use the documentation provided by your tape management vendor for more specifics.

Figure 58 shows page 4 of the ACS Test Case Define Panel:

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                    ACS TEST CASE VOLUME SERIAL SPECIFICATION       Page 4 of 4
Command ===>


ACS Test Library  : SCDS1.ACSTEST.LIB
ACS Test Member . : TEST1

To Complete ACS Test Case DEFINE, Specify:
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>         ===>         ===>
  Volser ===>         ===>         ===>

Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 58. Creating ACS Test Cases, Page 4 of 4

On these four panels, you can specify test values that correspond to the ACS variables listed in “Chapter 15. Writing ACS Routines” on page 265. Use the END command to save your test values and return to the ACS Test Selection panel. In the Test Selection panel, you can add more test case members to the same test library, or you can build another test library.

Running ACS Test Cases

To run the test cases in a test library, specify the library name and select option 3 in the ACS Test Selection panel (Figure 54 on page 149). You get the Test ACS Routines panel shown in Figure 59 on page 152.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                                TEST ACS ROUTINES
Command ===>


To Perform ACS Testing, Specify:

  CDS Name . . . . . . 'SMS.SCDS1.SCDS'
                                (1 to 44 Character Data Set Name or 'Active')
  ACS Test Library . . 'SCDS1.ACSTEST.LIB'
  ACS Test Member  . . TEST1     (fully or partially specified or * for all
                                members)
  Listing Data Set . . 'TEST1.LISTING'
                                (1 to 44 Character Data Set Name or Blank)

Select which ACS Routines to Test:

  DC ===> Y  (Y/N)  SC ===> Y  (Y/N)  MC ===> Y  (Y/N)  SG ===> Y  (Y/N)




Use ENTER to Perform Verification and Testing;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 59. Testing an ACS Routine

On this panel, you indicate which routines you want to test. Specifying an asterisk in the ACS Test Member field is convenient for running all test cases in the partitioned data set library. You can create a listing of the results of the testing by specifying a data set name in the Listing Data Set field. Leaving this field blank prevents the creation of a listing.

You can now test the routines by pressing Enter. If you specified a listing data set, the results of the testing are displayed in the PDF Browse panel shown in Figure 60.

```
 BROWSE -- TEST1.LISTING ---------------------------- LINE 00000000 COL 001 080
 COMMAND ===>                                            SCROLL ===> PAGE
********************************* TOP OF DATA  *********************************
                                ACS TESTING RESULTS

CDS NAME         : 'SMS.SCDS1.SCDS'

ACS ROUTINE TYPES: SG

ACS TEST LIBRARY : SCDS1.ACSTEST.LIB

  ACS TEST
  MEMBER       EXIT CODE   RESULTS
  ---------    ----------  ------------------------------------
  TEST1                 0  SG = DATABASE


ACS TESTING RC:  00
******************************** BOTTOM OF DATA ********************************
```

Figure 60. Results of Testing an ACS Routine

After examining the results, issue the END command and specify on the ACS Output Listing Disposition whether or not to keep the output listing.

ACS Routines Invoked for Copying and Importing Data Sets

Table 9 shows which ACS routines are invoked when performing initial allocation, importing, copying, restoring and recalling data sets.

Table 9. Allocation, IMPORT, and COPY Conditions

Type of Processing	Data Class ACS	Storage Class ACS	Management Class ACS	Storage Group ACS
Initial Allocation	Yes	Yes	SC	SC
IMPORT (Access Method Services)	No	Yes	SC	SC
COPY (DFSMSdss)	No	Yes	SC	SC
COPY BYPASSACS (DFSMSdss)	No	No	No	SC

Note:
Yes = ACS routine is invoked
No = ACS routine is not invoked (The ACS routine is not invoked for the data set that is being copied or imported as their attributes are already defined. The ACS routine might be invoked for other new data sets allocated to the job.)
SC = ACS routine is invoked only if storage class is assigned

Initial Allocation

Because the data set is new in an initial allocation, the data class ACS routine is invoked to determine the characteristics of the data set. The storage class ACS routine is then invoked to determine whether the data set is system-managed.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed and a storage class was assigned.

Each time a dynamic allocation is issued by application programs, such as DFSMShsm RECALL, SMS follows this initial allocation path (except for the data class ACS routine which is not invoked).

IMPORT (Access Method Services)

The IMPORT (IDCAMS) command has three conditions:

- The VSAM data set already exists
- The VSAM data set is preallocated (allocated but empty)
- The target data set is created by the IDCAMS IMPORT job

The data is copied from a backup copy or another VSAM data set, and is imported to the target data set. If a storage class is assigned, the data is then copied into a system-managed target data set. Otherwise it is copied into a non-system-managed target data set. The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the IDCAMS IMPORT job.

The storage class ACS routine is invoked to determine whether the data set is system-managed. The imported data might come from a non-system-managed source data set that was then copied into a system-managed target data set.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed and a storage class was assigned.

COPY (DFSMSdss)

The COPY command has three conditions:

- The target data set already exists
- The target data is preallocated (allocated but empty)
- The target data is created by the DFSMSdss COPY job

The data is either copied from a backup copy of the data set, or from another data set to the target data set.

The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the DFSMSdss COPY job.

The storage class ACS routine is invoked to determine whether the data set is system-managed. The copied data might come from a non-system-managed source data set that was then copied to a system-managed target data set.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed.

See *z/OS DFSMSdss Storage Administration Guide* for more information on the DFSMSdss COPY command.

COPY BYPASSACS (DFSMSdss)

The COPY BYPASSACS command has three conditions:

- The target data set already exists
- The target data is preallocated (allocated but empty)
- The target data is created by the DFSMSdss COPY job

The data is copied from either a backup copy or another data set to the target data set. If a storage class is assigned, the data is then copied into a system-managed target data set. Otherwise it is copied into a non-system-managed target data set.

The data class, storage class, and management class ACS routines are not invoked because the BYPASSACS keyword is coded in the DFSMSdss COPY job.

The storage class ACS routine is invoked only if it is specified in the DFSMSdss COPY job.

The storage group ACS routine is invoked only if SC is specified in the DFSMSdss COPY job, or if the data set is system-managed.

Requirement: You need RACF authorization to use the BYPASSACS keyword. The storage or security administrator must define BYPASSACS to RACF as a facility and then tell RACF who is authorized to use it.

ACS Routines Invoked for Restoring, Recalling, Recovering, and Converting Data Sets

Table 10 on page 155 shows which ACS routines are invoked for restoring, recalling, recovering, and converting data sets.

Table 10. RESTORE, RECALL, RECOVER, CONVERTV, and FORCENONSMS Conditions

Type of Processing	Data Class ACS	Storage Class ACS	Management Class ACS	Storage Group ACS
DFSMSdss RESTORE	No	Yes	SC	SC
DFSMSdss RESTORE BYPASSACS	No	No	No	SC
DFSMSdss CONVERTV	No	Yes	SC	No
DFSMShsm RECALL/ RECOVER	No	Yes	SC	SC
DFSMShsm FORCENONSMS	No	No	No	No

Note:
Yes = ACS routine is invoked
No = ACS routine is not invoked (The ACS routine is not invoked for the data sets as their attributes are already defined. The ACS routine might be invoked for other new data sets allocated to the job.)
SC = ACS routine is invoked only if storage class is assigned

DFSMSdss RESTORE

When using the DFSMSdss RESTORE command, the data set either exists or is deleted. The data set is recovered by restoring data from a backup copy to the target data set.

The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the backup data set or DFSMSdss RESTORE job.

The storage class ACS routine is invoked to determine whether the data set is system-managed.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed and a storage class is assigned.

DFSMSdss RESTORE BYPASSACS

When using the DFSMSdss RESTORE BYPASSACS command, the data set either exists or has been deleted. The data set is recovered by restoring data from a backup copy to the target data set.

The data class, storage class, and management class ACS routines are not invoked because the BYPASSACS keyword is coded in the DFSMSdss RESTORE job.

The storage group ACS routine is only invoked if:
- Storage class is specified in the DFSMSdss RESTORE job.
- The data set is system-managed.

Requirement: You need RACF authorization to use the BYPASSACS keyword. Define BYPASSACS to RACF as a facility and then tell RACF who is authorized to use it.

DFSMSdss CONVERTV

When using the DFSMSdss CONVERTV command, the data class ACS routine is not invoked because the data sets on the volume already exist.

The storage class ACS routine is invoked because the data sets and volume are to become system-managed.

The management class ACS routine is invoked, if the storage class is assigned, to determine the appropriate management class names assigned to the data sets.

The storage group ACS routine is not invoked because all the data sets on the system-managed volume are assigned to the storage group to which the volume belongs.

DFSMShsm RECALL/RECOVER

When using the RECALL/RECOVER command, if you want to RECALL a data set, then the data set already exists. If you want to RECOVER a data set, then either the data set exists or has been deleted. The data set is recovered by restoring data from a DFSMShsm backup copy to the target data set.

The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the backup copy.

The storage class ACS routine is invoked to determine whether the data set is system-managed. The restored data might come from a non-system-managed backup copy, and is allocated as system-managed or non-system-managed as determined by the ACS routines.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed.

DFSMShsm FORCENONSMS

When using the FORCENONSMS keyword, if you want to RECALL a data set, then the data set already exists. If you want to RECOVER a data set, then either the data set exists or is deleted. The data set is recovered by restoring data from a DFSMShsm backup copy to the target data set. Because the data set is not supposed to be system-managed, no ACS routines are invoked.

ACS Routine Environments

Depending on the environment, SMS invokes some or all of the ACS routines in the following order:

1. Data class
2. Storage class
3. Management class
4. 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 MSPOOL and MSPOLICY read-only variables if the pre-ACS installation exit has not done so. However, DFSMSrmm does not use the installation exit.

See *z/OS DFSMSrmm Implementation and Customization Guide* for detailed information on the DFSMSrmm support for the SMS pre-ACS interface. Or, check with your tape management vendors for information on support for the variables.

JCL DD Statement (Batch), and Dynamic Allocation

For the &ACSENVIR='ALLOC' environment, the data class ACS routine and then the storage class ACS routine are executed. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

Volume Reference

You can code VOL=REF on a DD statement to refer to a DD statement in the same or an earlier step to allocate the data set on the same volume as the earlier data set. You can also code VOL=REF to an existing data set (VOL=REF=A.B.C,...) where A.B.C is a cataloged data set. If VOL=REF is coded, SMS invokes ACS routines as follows:

- SMS invokes the data class ACS routine.
- Storage class is copied from referenced data set if the referenced data set has a storage class assigned to it.

  Notes:

  1. This is not true for non-SMS data sets, since they do not have storage classes to copy. In these cases, the routine is invoked and can either allow the non-SMS allocation or fail it. If it does anything else, SMS fails it.
  2. Storage class is copied from a referenced data set if the referenced data set has a storage class assigned to it, otherwise the storage class ACS routine is run to determine a storage class.

     For example, in case of VOL=REF to a SMS-managed tape date set, the storage class ACS routine must use the following read-only variables: &LIBNAME, &ANYVOL, &ALLVOL to derive a valid storage class.
  3. Because of the tendency to copy JCL, the VOL=REF to a non-SMS-managed data set might be used when the data set is allocated as old (DISP=OLD or DISP=SHR). Since the VOL=REF is used in locating an existing data set, allowing the data set to be allocated as SMS-managed when it is new could cause problems at this time.
  4. 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.
- Management class ACS routine is called if storage class is not null.
- The storage group from the referenced data set is passed as input to the storage group ACS routine. For data sets on SMS-managed tape volumes, the ACS routine must assign the same storage group to the referencing data set. For other SMS-managed data sets, any pool or VIO storage group can be assigned to the referencing data set.

When VOL=REF is used, the &ALLVOL and &ANYVOL ACS read-only variables are set to ’REF=SD,’ ’REF=ST’ or ’REF=NS’ as appropriate. Additionally, if the reference is to a data set on an SMS-managed volume, the storage group of the referenced data set is provided in the &STORGRP read-write variable, if it is available. (For some references to data sets on SMS-managed tape, it might not be, in which case the ACS routine should use the value of the &LIBNAME read-only variable instead.) If the referenced data set is new, it can still have multiple candidate storage groups since it hasn’t been allocated yet. In that case only the *first* candidate storage group is passed in.

The &ALLVOL and &ANYVOL ACS read-only variables contain the following values when you use VOL=REF:

- ’REF=SD’ (the volume reference is to an SMS-managed DASD or VIO data set)
- ’REF=ST’ (the volume reference is to an SMS-managed tape data set)

  Figure 61 illustrates these values:

```
PROC STORGRP

SELECT(&ANYVOL)
  WHEN('REF=SD')
    IF &DSTYPE = 'TEMP' & &&DSORG ^= 'VS' THEN
      SET &STORGRP = 'VIOSG','MAIN3380','MAIN3390','SPIL3380','SPIL3390'
    ELSE
      SET &STORGRP = 'MAIN3380','MAIN3390','SPIL3380','SPIL3390'
  WHEN('REF=ST')
    SET &STORGRP = &STORGRP
  OTHERWISE
  .
  .
  .
END
END
```

Figure 61. Example of REF=ST Values when Using VOL=REF

- ’REF=NS’ (the volume references is to a non-SMS-managed data set)

  Figure 62 illustrates these values:

```
PROC 0 STORCLAS
FILTLIST AUTH_USER INCLUDE('SYSPROG1','SYSPROG2','STGADMIN','SYSADMIN')

IF &ANYVOL = 'REF=NS' & &HLQ ^= 'SYS1' THEN
  IF &USER ^= &AUTH_USER THEN
    DO
      WRITE 'INVALID USE OF VOL=REF TO A NON-SMS-MANAGED DATA SET'
      WRITE 'AS OF 12/31/94 ALLOCATION WILL BE FAILED FOR ' &DSN
    END
  .
  .
  .
END
```

Figure 62. Example of REF=NS Values when Using VOL=REF

As of the date shown in Figure 62 on page 158, you might change the ACS routine to fail the uses of VOL=REF that are not valid, as shown in Figure 63.

```
PROC 1 STORCLAS
FILTLIST AUTH_USER INCLUDE('SYSPROG1','SYSPROG2','STGADMIN','SYSADMIN')

IF &ANYVOL = 'REF=NS' & &HLQ ^= 'SYS1' THEN
  IF &USER ^= &AUTH_USER THEN
    DO
      WRITE 'INVALID USE OF VOL=REF TO A NON-SMS-MANAGED DATA SET'
      WRITE 'DATA SET ' &DSN ' MUST BE SMS-MANAGED'
      EXIT CODE(4)
    END
  .
  .
  .
END
```

Figure 63. Example of Failing VOL=REF Values That Are Not Valid

Data Set Stacking

Data set stacking is the function used to place several data sets on the same tape volume or set of tape volumes. It increases efficiency when using tape media and reduces the overall number of tape volumes needed by allocation. It also allows an installation to group related data offsite. The data set sequence number subparameter on the JCL LABEL parameter is used in conjunction with VOL=REF or VOL=SER to accomplish this function.

Under certain conditions, SMS invokes the ACS routines more than once. This section describes those conditions as well as the values of the read-only variables for the initial invocation and any subsequent invocations of the ACS routines. 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.

Using VOL=SER Within a Job Step

When data set stacking is specified using VOL=SER within a job step, the system ensures that all the data sets making up the data set collection (a group of data sets intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking) are directed to the same device category. If the ACS routines initially directed the stacked allocations to different device categories, the system detects this and reinvokes the ACS routines, passing additional information to those routines. The ACS routines can then do one of the following:

- Correct the problem and route the allocations to consistent device categories
- Fail the stacked allocation (if the ACS routine exits with a nonzero return code)
- Fail to correct the inconsistency, in which case SMS fails the allocation

Recommendation: The system cannot detect data set stacking if VOL=SER is used to stack data sets across jobs or job steps. In these instances, you can change your JCL to specify VOL=REF instead of VOL=SER. For more recommendations on when to use VOL=REF versus VOL=SER for data set stacking, see *z/OS MVS JCL User's Guide*.

The system reinvokes 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, and a VOL=SER, where at least one of the volume serial numbers matches one of the volume serial numbers for a previous request in the same step.
- 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).

Using VOL=REF

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.

Possible Values for the &UNIT Read-Only Variable

Information is passed to the ACS routines in the &UNIT ACS read-only variable, so that the ACS routines know when data set stacking or unit affinity is used. 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.

Table 11 shows the values to which the &UNIT read-only variable can be set. The first value is applicable only when UNIT=AFF is used. The others are applicable only when data set stacking is being done, whether or not the UNIT=AFF keyword is present.

Table 11. 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
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

Note: ACS routines can be invoked three times in a JES3 environment.

Figure 64 shows an example of a storage class ACS routine for read-only variables:

```
PROC  &STORCLAS

SELECT(&UNIT)
  WHEN('STK=SMSD')
    SET &STORCLAS = 'POOLSC'
  WHEN('STK=NSMS')
    SET &STORCLAS = ''
  OTHERWISE
```

Figure 64. Example of a Storage Class ACS Routine for Read-Only Variables

The storage group ACS routine could then do something like what is shown in Figure 65:

```
PROC  &STORGRP

SELECT(&UNIT)
  WHEN('STK=SMSD')
    SET &STORGRP = 'S1P03'
  WHEN('STK=NSMS')
    SET &STORGRP = 'POOLSG'
  OTHERWISE
```

Figure 65. Example of a Storage Group ACS Routine for Read-Only Variables

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 are 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 12 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 12. 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
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

Volume Reference and Unit Affinity

When both VOL=REF and UNIT=AFF are specified, VOL=REF processing takes precedence over UNIT=AFF. The &UNIT read-only variable may contain blanks or 'AFF= ', depending on the disposition of the referenced DD.

Recommendation: When specifying both VOL=REF and UNIT=AFF, ensure that the ACS routines check for &ANYVOL='REF=ST', 'REF=SD' or 'REF=NS' *before* checking the &UNIT read-only variable. This ensures that the ACS routines select the correct classes.

Unlike volume reference processing, in which the storage class is produced from the REF'ed DD, unit affinity processing allows you to break the affinity by setting a different storage class. Even though the AFF'ed DD's storage class is passed as input to the storage class routine, the storage class routine must set a valid storage class for the job to run successfully.

Example: In Figure 66, the storage class routine ensures that even if only UNIT=AFF is specified, the AFF'ed DD storage class will be assigned.

```
IF &ANYVOL = 'REF=ST' OR &UNIT = 'AFF=SMST' THEN
  DO
    IF &STORCLAS = '' THEN
       SET &STORCLAS = &STORCLAS      /* TO HONOR AFFINITY */
    ELSE
       SET &STORCLAS = 'XXXXXXXX'        /* TO BREAK AFFINITY */
    WRITE 'STORCLAS SET TO &STORCLAS'
    EXIT CODE(0)
  END
ELSE
  IF &ANYVOL = 'REF=SD' OR &UNIT = 'AFF=SMSD' THEN
    DO
      SET &STORCLAS = &STORCLAS
WRITE 'STORCLAS SET TO &STORCLAS'
      EXIT CODE(0)
    END
```

Figure 66. Example of a Storage Class ACS Routine Specifying Both Volume Reference and Unit Affinity

Tape Management System Support

To support your tape management system's need to coordinate complex vaulting requirements with data set allocation in system-managed environments, you can use a pre-ACS routine exit to set values for read-only variables which SMS then uses as input to ACS routines. This support is useful when the choice of a storage group needs to be influenced by the data set's vaulting requirements. For example, you might use this support when excluding data from redirection under tape mount management or directing data to a specific system-managed library.

The four read-only variables used to support tape management system-driven tape allocations are:

&MSPDEST used to specify a destination, in data set name format. This format lets you specify a sequence of destinations to be identified, where each qualifier is a specific destination. For example, a data set vaulted first at location OUTD and then sent to OLTS could have an MSPDEST of `'OUTD.OLTS'`. The actual values depend on the support provided by your tape management system.

&MSPARM used to specify any additional information related to a system-driven tape management allocation. This is a variable length field, and its value is specified through an external exit.

&MSPOLICY used to specify a management policy related to a system-driven tape management allocation. This is an 8 character field, and its value is specified through an external exit.

&MSPOOL used to specify a tape pool name associated with the data set being allocated. In a system-managed tape environment with scratch pool support, this variable might be used to specify a default storage group, where the tape storage group is equivalent to the tape pool specified in the variable.

DFSMSrmm allows you to move your pooling decisions to the SMS ACS routines so that you can assign a scratch pool based on ACS input variables. A scratch pool selected in this way equates to a storage group and includes all assigned volumes in the DFSMSrmm control data set. DFSMSrmm passes scratch pooling decisions using the pre-ACS interface in the MSPOOL variable so that it can be the base for assigning a storage group. For non-system-managed tapes, DFSMSrmm also calls the management class and storage group ACS routines so that a storage group can be assigned for use as a scratch pool name.

You can assign a storage group using the DFSMSrmm RMMPOOL environment call. DFSMSrmm uses the assigned storage group as the pool name to validate that a mounted scratch volume is from the requested pool.

DFSMSrmm also allows you to move your VRS policy assignments to the SMS routines so that you can use any of the existing ACS input variables as the base for assigning a management class. You can assign a VRS policy by name to a system-managed or non-system-managed tape data set, and DFSMSrmm uses this management class name to identify the VRS policy. DFSMSrmm passes VRS management value decisions using the pre-ACS interface in the MSPOLICY variable so that it can be the base for management class assignments. DFSMSrmm also calls ACS routines for assigning a management class to non-system-managed tapes.

When DFSMSrmm calls ACS routines for a management class or storage group name, the &ACSENVIR variable is set to RMMVRS for the required management class and RMMPOOL for the required storage group. When RMM calls the ACS routines with the &ACSENVIR variable set to either RMMPOOL or RMMVRS, &STORCLAS is set to USERMM.

See *z/OS DFSMSrmm Guide and Reference* and *z/OS DFSMSrmm Implementation and Customization Guide* for further information on the SMS pre-ACS interface and the updates to the ACS input variables.

Access Method Services

For the ALLOCATE and DEFINE commands (&ACSENVIR='ALLOC'), the data class ACS routine and then the storage class ACS routine are executed. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

For the IMPORT command, the storage class ACS routine is executed first. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

Redrive MGMTCLAS ACS routine on data set rename invokes the management class ACS routine when an SMS managed cluster, GDS or non-VSAM data set is renamed. Catalog management invokes MGMTCLAS ACS routines during RENAME processing. You can reassign a different management class based on the following data set attributes:

- New data set name
- Data set type
- Data set organization
- Expiration date in the catalog
- Old management class
- Data class
- Storage class
- Record organization
- User information
- Group information

ACS routines support RENAME. The following read-only variables are set for the management class ACS routine:

- &ACSENVIR:RENAME
- &DSN of the new data set name
- &DSORG
- &DSTYPE
- &EXPDT
- &STORCLAS
- &DATACLAS
- &MGMTCLAS
- &GROUP of the job
- &USER of the job
- &HLQ
- &LLQ

DFSMShsm

The management policy during a data set recall or recover is determined by the value set for &ACSENVIR in the management class ACS routine. The storage class ACS routine is then invoked to apply performance and availability criteria. The data class routine is not invoked. Additionally, during DFSMShsm RECALL processing, the storage group routine gets a different value for the &ACSENVIR field, depending on the data mover that is used. If DFSMSdss is the data mover, RECOVER is passed as the environment; if DFSMShsm is the data mover, RECALL is passed as the environment. Because of this, do not use the storage group ACS routine to test the value in &ACSENVIR for recall or recover, as this could yield inconsistent results.

DFSMSdss

For the COPY command (&ACSENVIR='ALLOC') and the RESTORE command (&ACSENVIR='RECOVER'), the storage class ACS routine is executed first. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

For CONVERTV SMS TEST and CONVERTV SMS, (&ACSENVIR='CONVERT'), the storage class ACS routine is executed first. If the storage class ACS routine determines that the storage class is not null, the management class ACS routine is executed.

DFSMSrmm

DFSMSrmm calls the ACS routines to request the assignment of storage group and management class names for non-system-managed tape data sets. Table 13 lists the read-only variables that are set for DFSMSrmm requests.

Table 13. DFSMSrmm Read-Only Variables

Variable	RMMPOOL Environment	RMMVRS Environment
&UNIT	Yes	Yes
&USER	Yes	Yes
&GROUP	Yes	Yes
&DSORG	Yes	Yes
&DSTYPE	Yes	Yes
&XMODE	Yes	Yes
&JOB	Yes	Yes
&DD	Optional for mount messages	Yes
&PGM	Optional for mount messages	Yes
&EXPDT	Optional for mount messages	Yes
&SYSNAME	Yes	Yes
&SYSPLEX	Yes	Yes
&STORGRP	Yes Pool prefix or name	Yes
&ACSENVIR	Yes RMMPOOL	Yes RMMVRS
&DSN	Yes	Yes
&HLQ	Yes	Yes
&LLQ	Yes	Yes
&NQUAL	Yes	Yes
&ACCT_JOB	Yes	Yes
&ACCT_STEP	Yes	Yes
&FILENUM	Optional for mount messages	Yes
&LABEL	Yes	Yes
&LIBNAME	Optional for JES3 fetch messages	Yes

When the &ACSENVIR variable is set to RMMPOOL, DFSMSrmm requests you to return a storage group name. DFSMSrmm requests that both the management class and storage group ACS routines are run. With a combination of these routines, you can decide whether or not you want to return a storage group value and what the value will be. If a storage group name is returned, it must be a valid tape storage group name. Using different &ACSENVIR values helps you differentiate between DFSMSrmm requests for a storage group name and allocation requests for system-managed data sets.

When the &ACSENVIR variable is set to RMMVRS, DFSMSrmm requests you to return a management class name. DFSMSrmm requests that only the management class ACS routine is run. Using different &ACSENVIR values helps you differentiate between DFSMSrmm requests for a management class name and allocation requests for system-managed data sets.

ISMF

When you are testing ACS routines, the data class ACS routine and then the storage class ACS routine are executed. If the storage class ACS routine determines that the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed. You can also execute each ACS routine separately from the others when performing tests.

OAM

For the OSREQ CHANGE command (&ACSENVIR='CHANGE'):

- If storage class, or both storage class and management class, are specified, both the storage class and management class ACS routines are executed, in that order.
- If only management class is specified, only the management class ACS routine is executed, with the old storage class used as input.

For the OSREQ STORE command (&ACSENVIR='STORE'):

- If the object is the first object in a collection, the storage class, management class and storage group ACS routines are executed, in that order. These routines determine the defaults for the object collection.
- Whenever storage class or management class is specified, the storage class and management class ACS routines are executed for the object with the specified classes as input. Then the ACS routines derive the initial storage class and management class for the object rather than having the object use the default initial storage and management classes for the collection.

At class transition time, as defined by the management class associated with an object, (&ACSENVIR='CTRANS'), the storage class routine and then the management class routine are executed.

Processing of SMS Classes and Storage Groups

For each of the SMS classes, the processing is as follows:

1. If you have an ACS routine in your CDS to determine the SMS class, SMS executes the routine.
2. Next, SMS executes the corresponding *ACS installation exit*. An ACS installation exit is an assembler language program you can write to perform processing beyond the scope of the standard ACS routines. Such processing might involve:
   - Calling other programs
   - Writing SMF records
   - Writing generalized trace facility (GTF) trace records
   - Performing arithmetic calculations
   - Maintaining large tables of information for quick searches
   - Writing dumps
   - Invoking ACS routines only once

Each exit can override the corresponding SMS class, whether explicitly specified or previously determined by an ACS routine. The exit can also invoke the corresponding ACS routine a second time, but this does not cause the installation exit to be reinvoked.

See *z/OS DFSMS Installation Exits* for additional information about ACS installation exits.

3. Finally, if RACF is installed and the storage administrator has defined STORCLAS and MGMTCLAS as resources and permitted end users to specify them, the system verifies that the class is defined in the currently active configuration. The system then checks to verify that end users are allowed to use a selected management class or storage class.

For storage groups, SMS invokes only the storage group ACS routine. Storage group does not have a corresponding ACS installation exit.

Displaying ACS Object Information

To display information about the ACS objects stored in a control data set, select option 5, DISPLAY, from the Automatic Class Selection Application Selection panel and press Enter. You receive the ACS Object Display panel shown in Figure 67.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                              ACS OBJECT DISPLAY
Command ===>


CDS Name  : SMS.SCDS1.SCDS

ACS Rtn   Source Data Set ACS      Member    Last Trans  Last Date  Last Time
Type      Routine Translated from  Name      Userid      Translated Translated
--------  -----------------------  --------  ----------  ---------- ----------
DATACLAS  STAGE2.SYSTEM1.ACSRTN.   DATACLAS  IBMUSER     1994/08/24  15:27
          SOURCE

MGMTCLAS  STAGE2.SYSTEM1.ACSRTN.   MGMTCLAS  IBMUSER     1993/11/04  16:53
          SOURCE

STORCLAS  STAGE2.SYSTEM1.ACSRTN.   STORCLAS  IBMUSER     1995/02/15  15:58
          SOURCE

STORGRP   STAGE2.SYSTEM1.ACSRTN.   STORGRP   IBMUSER     1993/11/04  16:53
          SOURCE


Use HELP Command for Help; Use END Command to Exit.
```

Figure 67. Displaying ACS Object Information

CDS NAME can be the name of the SCDS that you entered on the Automatic Class Selection Application Selection panel or '`ACTIVE`'. It identifies the data set that contains the ACS objects that are to be displayed.

ACS RTN TYPE always lists the four types of ACS objects that can belong to an SCDS, even if they do not actually exist in the SCDS being displayed. If they do not exist, the corresponding display fields contain dashes.

SOURCE DATA SET ACS ROUTINE TRANSLATED FROM contains the name of the data set that has the source code responsible for creating the ACS object. The figure shows that the data set name is folded if it is more than 23 characters in length. If an ACS object of a particular type does not exist in the SCDS being displayed, then this field contains dashes.

MEMBER NAME displays the member name within the data set that contains the ACS source code used to create the corresponding ACS object. If an ACS object of the particular type does not exist in the SCDS being displayed, or if the SOURCE DATA SET ACS ROUTINE TRANSLATED FROM is not a PDS or PDSE, the field contains dashes.

LAST TRANS USERID displays the TSO user ID of the person who last translated the ACS routine. If an ACS object of the particular type does not exist, the field contains blanks.

LAST DATE TRANSLATED displays the date when the corresponding ACS object was created. If an ACS object of the particular type does not exist, the field contains blanks.

LAST TIME TRANSLATED displays the time when the corresponding ACS object was created. If an ACS object of the particular type does not exist, the field contains blanks.

Deleting an ACS Object from an SCDS

To delete an ACS object from an SCDS, select option 6, DELETE, from the Automatic Class Selection Application Selection panel and press Enter. You receive the Delete ACS Object panel shown in Figure 68.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                              DELETE ACS OBJECT
Command ===>


To Delete an ACS Object, Specify:

  SCDS Name  . . . . . 'SMS.SCDS1.SCDS'
                          (1 to 44 Character Data Set Name)

  ACS Routine Type . . SG  (DC=Data Class, MC=Management Class, SC=Storage
                          Class, SG=Storage Group)









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

Figure 68. Deleting ACS Objects

The SCDS Name field contains the name of the SCDS from which the ACS object is to be deleted. It is a required field and is primed with the last referenced SCDS. The SCDS name cannot be ’ACTIVE’. The default is blank.

The ACS Routine Type field contains the type of ACS object. It is a required field and it is primed with the last used value. The default is blank.

When you press Enter, you receive the Confirm Delete Request panel to certify that you want to delete the ACS object.

Chapter 10. Activating Storage Management Subsystem Configurations

You can activate an SMS configuration manually, or automatically at IPL. This chapter shows you how to perform the initial activation of an SMS configuration using a four-step manual approach. It also explains how you can activate an SMS configuration automatically at IPLs. In addition, it explains how you can change individual SMS parameters with the SETSMS operator command.

Prerequisite: When you activate an SMS configuration, ensure that all of the DASD volumes that belong to the configuration are initialized as SMS volumes. Otherwise, attempted allocations to an improperly initialized volume will fail. However, initialization for tape volumes is no different for SMS-managed and non-SMS-managed volumes.

Manually Activating the First Storage Management Subsystem Configuration

IGDSSIIN is the subsystem initialization routine module for SMS. By omitting it from the SMS entry of IEFSSNxx for each system in the SMS complex, you can manually control the activation of an SMS configuration. Refer to “Chapter 2. Preparing for the Storage Management Subsystem” on page 9.

Step One: IPL Each System in the SMS Complex

After defining SMS as a subsystem to z/OS, IPL each system in the SMS complex. The presence of the SMS entry in IEFSSNxx tells z/OS to recognize SMS as a valid subsystem within each system. The absence of the IGDSSIIN module name in IEFSSNxx tells the system that you want to manually start SMS.

Step Two: Prepare One System

From one system in the SMS complex, issue the T SMS=*xx* command, where *xx* identifies IGDSMSxx as the SMS initialization control member of SYS1.PARMLIB. SMS uses the ACDS and COMMDS identified in IGDSMSxx to manage storage. Because the initial ACDS and COMMDS are empty, the system is activated with a *null configuration*. Keep in mind that a null configuration is only intended as a migration path.

For a complete list of the IGDSMSxx member parameters, see Table 1 on page 15.

Requirement: All systems in the SMS complex must be running in the same mode. When an SMS control data set that supports only eight names is accessed for update on a system running in 32-name mode, you must convert the data set to a new, incompatible format in order to support 32 names. Confirm this conversion using the operator console or the ISMF. This conversion is permanent, so make copies of your control data sets before the system mode is converted.

Step Three: Activate the Configuration from One System

The configuration is only activated once for the SMS complex. It is not necessary to activate a configuration from every system in the SMS complex. After activating SMS with a null configuration, activate an SMS configuration contained in a valid

© Copyright IBM Corp. 1986, 2002

SCDS on the same system. You can use either the ISMF ACTIVATE command or the SETSMS operator command. Both procedures copy the contents of the SCDS to the ACDS specified in IGDSMSxx.

When an SMS control data set that supports only eight names is accessed for update on a system running in 32-name mode, you must convert the data set to a new, incompatible format in order to support 32 names. Confirm this conversion, using the operator console or ISMF. This conversion is permanent, so you should make copies of your control data sets before the system mode is converted.

Activating with the ISMF ACTIVATE Command

On the Control Data Application Selection panel shown in Figure 5 on page 28, specify the name of an SCDS and issue the ACTIVATE command from the command line. A *Write to Programmer* message indicates if the activation is successful, provided you have WTPMSG in the TSO/E PROFILE.

Activating with the SETSMS Operator Command

From the operator console, issue the command:

```
SETSMS SCDS(dsname)
```

where `dsname` identifies the name of the SCDS to be activated. Using the SCDS defined earlier in this manual, the command would be:

```
SETSMS SCDS(SMS.SCDS1.SCDS)
```

Step Four: Activate SMS on the Other Systems

For the other systems in the SMS complex, use the T SMS=xx command to start SMS on those systems, using the SMS configuration identified in Step Three. In each system, IGDSMSxx specifies the name of the ACDS containing the SMS configuration. All of the IGDSMSxx members must point to the same ACDS. Because the ACDS is no longer empty, the systems use it (and the COMMDS) to manage storage.

Automatically Activating a Storage Management Subsystem Configuration

For each system in the SMS complex, update the SMS entry in IEFSSNxx to include the IGDSSIIN module name.

Make certain that each ID field identifies the IGDSMSxx member containing the name of the last used ACDS. All of the IGDSMSxx members must point to the same ACDS. At future IPLs, the SMS configuration contained in the ACDS is activated by all systems in the SMS complex. For example:

```
SMS,IGDSSIIN,'ID=02,PROMPT=DISPLAY'
```

indicates that the ACDS specified in IGDSMS02 contains the SMS configuration to be activated at future IPLs.

Changing Storage Management Subsystem Parameters

Once you have activated an SMS configuration, you can use the `SETSMS` operator command to change SMS parameters.

Restriction: Some of the parameters contained in the IGDSMSxx parmlib member cannot be changed using the SETSMS operator command. The section "Parameters of the SETSMS Operator Command" lists only those parameters that can be changed using the SETSMS operator command. For a complete list of the IGDSMSxx parmlib member parameters, see Table 1 on page 15.

Parameters of the SETSMS Operator Command

The following parameters can be changed using the `SETSMS` operator command.

See Also: For more detailed descriptions of the parameters, see *z/OS MVS System Commands* and *z/OS MVS Initialization and Tuning Guide*.

ACDS(*dsname***)**
identifies the name of the data set containing the active configuration. If you omit *dsname*, the operator is prompted for a value.

ASID({*asid***|*}})**
limits tracing to a certain address space or permits it for all address spaces. Specify * if you want to trace all address spaces (providing SMS tracing is active). This is the default. You can enter up to 4 digits for the ASID keyword. If you leave off the leading zeroes, they are inserted.

BMFTIME({*nnn***|3600})**
specifies the number of seconds between recording SMF record type 42, subtype 1 records for PDSE (where the buffer hits are I/O requests which did not require actual disk I/O). You can specify a value from 1 to 86399 (23 hours, 59 minutes, 59 seconds), and the default is 3600 (1 hour).

CACHETIME({*nnn***|3600})**
specifies the number of seconds between recording SMF records for device cache use. The **CACHETIME** parameter applies only to the volumes behind an IBM 3990 Storage Control with cache unit. You can specify a value from 1 to 86399 (23 hours, 59 minutes, 59 seconds), and the default is 3600 (1 hour).

CF_TIME({*nnn***|3600})**
indicates the number of seconds between recording SMF records for the coupling facility (both cache and lock). You can specify a value from 1 to 86399 (23 hours, 59 minutes, 59 seconds), and the default is 3600 (1 hour). This keyword sets the interval time for the following SMF 42 subtypes:

SUBTYPE 15	Coupling facility storage class average response time
SUBTYPE 16	Coupling facility data set average response time
SUBTYPE 17	Coupling facility lock structure activity
SUBTYPE 18	Coupling facility cache partition summary
SUBTYPE 19	SMSVSAM least recently used statistics summary

COMMDS(*dsname***)**
identifies the name of the COMMDS. If you omit *dsname*, the operator is prompted for a value.

DEADLOCK_DETECTION({*iiii*|15,*kkkk*|4})
specifies the deadlock detection intervals used by the DFSMSdfp Storage Management Locking Services. The first subparameter is the local deadlock detection cycle and specifies the interval in seconds for detecting deadlocks within a system. The second subparameter is the global deadlock detection cycle and specifies the interval for detecting deadlocks between systems. The value is specified as the number of local detection cycles that occur before global deadlock detection is initiated.

iiii one to four digit numeric value in the range 1-9999 that specifies the length in seconds of the local deadlock detection interval. The default is 15 seconds.

kkkk one to four digit numeric value in the range 1-9999 that specifies the number of local deadlock cycles that must complete before global deadlock detection occurs. The default is 4.

DESELECT({*event[,event][,...]*|ALL})
deletes items from the list of events to be traced (if SMS tracing is active). DESELECT has no default. If you specify events that conflict in SELECT and DESELECT, the keyword that appears last has final authority.

The events that you can specify on SELECT and DESELECT are:

MODULE SMS module entry or exit

DSTACK SMS Data Set Stacking Service

SMSSJF SMS/SJF interfaces

SMSSSI SMS/SSI interfaces

ACSINT ACS services interfaces

OPCMD Operator commands

CONFC Configuration change

CDSC Control data set changes

CONFS SMS Configuration services

MSG SMS Message services

ERR SMS Error recovery and recording services

CONFR Return data from an active configuration

CONFA Activate a new configuration

ACSPRO Perform ACS processing

IDAX SMS interpreter/dynamic allocation

DISP SMS disposition processing exit

CATG SMS catalog services

VOLREF SMS VOLREF services

SCHEDP SMS scheduling services (preallocate catalog orientation)

SCHEDS SMS scheduling services (system select)

VTOCL SMS VTOC/data set services (allocate existing data set)

VTOCD SMS VTOC/data set services (delete existing data set)

VTOCR SMS VTOC/data set services (rename existing data set)

VTOCC	SMS VTOC/data set services (allocate new data set)
VTOCA	SMS VTOC/data set services (add a volume to a data set)
RCD	SMS Recording services
DCF	SMS device control facility
DPN	SMS device pool name select subsystem interface
TVR	SMS tape volume record update facility
ALL	All of the above options

DINTERVAL({*nnn*|150})
specifies the number of seconds SMS allows to elapse before it reads device statistics, which SMS uses to manage hardware usage and maximize efficiency. The **DINTERVAL** parameter applies only to the volumes behind an IBM 3990 Storage Control with cache unit. You can specify a value from 1 to 999 (16 minutes, 39 seconds), and the default is 150.

INTERVAL({*nnn*|15})
specifies the *synchronization time interval* of the system, which is the number of seconds between system checks of the COMMDS for news about SMS configuration changes from other systems in the SMS complex. You can specify a value from 1 to 999 (16 minutes, 39 seconds), and the default is 15.

Recommendation: OAM depends on the time interval specified with this INTERVAL keyword in order to be notified of an SCDS activation. Therefore, be sure to specify an appropriate value for this keyword when it is used by other address spaces that are also dependent on the SMS configuration.

JOBNAME({*jobname*|*})
limits tracing to a certain job or permits tracing on all jobs. The default is *****, to trace all jobs. This keyword supports objects or tape libraries.

RLSINIT({NO|YES})
specify YES if you want the SMSVSAM address space started as part of system initialization or the V SMS,SMSVSAM,ACTIVE command. This value applies only to the system accessed by the parmlib member and is acted upon when SMSVSAM is next started. The default is NO.

[RLS_MaxCfFeatureLevel({A|Z})]
specifies the method that VSAM RLS uses to determine the size of the data that is placed in the CF cache structure.

If you specify **A**, caching proceeds using the RLSCFCACHE keyword characteristics that are specified in the SMS data class that is defined for the VSAM sphere.

If you do not specify a value, or if you specify **Z**, then only VSAM RLS data that have a Control Interval (CI) value of 4K or less are placed in the CF cache structure. The default is **Z**.

Restrictions:

- If **A** is specified for the RLS_MaxCfFeatureLevel parameter, systems lower than V1R3 will not be able to connect to the CF cache structure.

- If a lower-level system is the first system activated in the sysplex, RLS_MaxCfFeatureLevel defaults to **Z**, and all systems will be able to connect to the CF cache structure.
- If the SETSMS command is used to change the RLS_MaxCfFeatureLevel value to **A** on a mixed-level system, the command is rejected and message IGW500I is issued.

RLS_MAX_POOL_SIZE({*nnnn***|100})**
specifies the maximum size in megabytes of the SMSVSAM local buffer pool. SMSVSAM attempts to not exceed the buffer pool size you specify, although more storage might be temporarily used. Because SMSVSAM manages buffer pool space dynamically, this value does not set a static size for the buffer pool.

Use SMF 42, subtype 19 records to help you determine the maximum size of the SMSVSAM local buffer pool.

You can specify a two to four-digit numeric value, with 10 as the minimum value. If you specify a value less than 10, the field is set to 10. If you specify a value greater than 1500, SMSVSAM assumes there is no maximum limit. We recommend that you limit the size of the local buffer pool.

The default is 100.

RLSTMOUT({nnn|0})
specifies the maximum time, in seconds, that a VSAM RLS request must wait for a required lock before the request is assumed to be in deadlock. You can specify a value between 0 to 9999 (in seconds). A value of 0 means that the VSAM RLS request has no time out value and the request will wait for as long as necessary to obtain the required lock.

RLSTMOUT can be specified only once in a sysplex and applies across all systems in the sysplex.

The default is 300.

SAVEACDS(*dsname***)**
specifies the ACDS that SMS uses to save the active configuration for backup or recovery purposes.

SELECT({*event[,event][,...]***|ALL})**
adds items to the list of events to be traced (if SMS tracing is active). The default is **ALL**. See **DESELECT** for a list of valid events.

SIZE(*nnn***{K|M})**
specifies the size of the SMS trace table in bytes. When the unit is **K**, the value can range from 0K to 255000K, and it is rounded up to the nearest 4K unit. When the unit is **M**, the value can range from 0M to 255M. If you specify a value of 0, no tracing is performed. The default is 128K.

SMF_TIME({YES|NO})
YES indicates that the following SMF type 42 records are created at the SMF interval time, and that all of the indicated records are synchronized with SMF and RMF data intervals:

SUBTYPE 1	Buffer management statistics
SUBTYPE 2	Cache control unit statistics (IBM 3990-3, 3990-6, IBM RVA, and IBM ESS)
SUBTYPE 15	Coupling facility storage class average response time

SUBTYPE 16	Coupling facility data set average response time
SUBTYPE 17	Coupling facility lock structure activity
SUBTYPE 18	Coupling facility cache partition summary
SUBTYPE 19	SMSVSAM least recently used statistics summary

DFSMS creates the specified SMF record at the end of the interval period and SMF sends the event notification signal. If YES is specified, this subparameter overrides the following subparameters: BMFTIME, CACHETIME, CF_TIME. The default is YES.

Related Reading: For more information on SMF, see *z/OS MVS System Management Facilities (SMF)*.

TRACE({OFF|ON})
indicates whether the SMS trace facility is activated. The default is **ON**. The SMS trace facility records trace records in the SMS address space.

See Also: For more information about the TRACE parameter, see *z/OS DFSMSdfp Diagnosis Reference*.

TRACEEXIT(*user_trace_exit***)**
specifies the name of your trace exit routine. You can specify an alphanumeric value of 1 to 8 characters, starting with an alphabetic character. Your value must represent a valid module name. The default value is blank. This interface is usually used only for diagnostic purposes under the guidance of the service organization and therefore is not documented here.

TYPE({ALL|ERROR})
specifies whether SMS traces all events or only errors. The default is **ERROR**.

If you do not specify a keyword value, SMS uses the current value (specified in IGDSMSxx or by a previous operator command).

Considerations when Changing Storage Management Subsystem Configurations

When activating a new SMS configuration, you have two options for keeping the currently active SMS configuration information:

- Keep, but never modify, the original SCDS from which the current SMS configuration was activated.

  If you choose this method, you need to maintain a log of all status changes, such as VARY storage group commands, that you make to the currently active SMS configuration. If in the future you activate a different SMS configuration but then decide you want to fall back to your original, you can reactivate the SCDS. You lose all the status changes you have made since activating the SCDS and must reenter them, but you return to the original SMS configuration.
- Save the current active SMS configuration using the SETSMS operator command:

  ```
  SETSMS SAVEACDS(ACDS.FALLBACK)
  ```

  This is the better alternative.

ACDS.FALLBACK must be an existing, already allocated data set. By using this command, you not only save the current storage management policy in ACDS.FALLBACK, but you also save the status changes you have made since the original SCDS was activated. You can then activate the new SMS configuration. If in the future you decide you want to fall back to the original SMS configuration, you can use the SETSMS operator command to reactivate it:

```
SETSMS ACDS(ACDS.FALLBACK)
```

This alternative is useful if you have altered the SCDS that you originally activated.

Restriction: Any SMS status that is changed using the SETSMS command is overridden by the MVS status if a new configuration is activated. See *z/OS MVS System Commands* for information on the SETSMS operator command.

Requirement: All systems in the SMS complex must be running in the same mode.

When an SMS control data set that supports only eight names is accessed for update on a system running in 32-name mode, you must convert the data set to a new, incompatible format in order to support 32 names. Confirm this conversion using the operator console or the ISMF. This conversion is permanent, so you should make copies of your control data sets before the system mode is converted.

OAM Considerations when Changing SCDSs

SMS notifies OAM when a new SCDS has been activated. OAM takes action depending on the RESTART parameter that is specified on the OAM address space. If RESTART=YES is specified, or defaulted, the OAM address space automatically restarts, rebuilding its configuration to match the newly activated SCDS. If RESTART=NO is specified, OAM does not automatically restart, but issues a message acknowledging that an activation has taken place. In this case, you must determine if an OAM restart is necessary. If a restart is necessary, use the MODIFY OAM,RESTART command.

When RESTART=YES is specified, the length of the delay between the SCDS activation and the OAM restart depends on the value specified in the INTERVAL keyword in the active IGDSMSxx PARMLIB member. During this reinitialization, all optical libraries and drives defined to the new SCDS are reset to the initial status values specified in the SCDS. After the OAM restart completes, display all optical libraries and drives, and tape libraries, and then set them to the desired online or offline status before the reinitialization occurred.

For more information, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Chapter 11. Maintaining the Storage Management Subsystem

After activating SMS, you need to monitor and adjust it over time. This chapter explains how to maintain SMS, such as using the DISPLAY SMS operator command. Maintenance activities include listing, altering, copying, and deleting SMS classes, aggregate groups, storage groups, optical libraries and drives, and tape libraries. They also include listing, altering and ejecting optical and tape volumes.

Tip: You can use the DFSMS NaviQuest tool to help you maintain your SMS configuration. With DFSMS NaviQuest, you can perform many ongoing storage administration activities in batch, thereby freeing the workstation for other work. For example, you can update configuration values, create reports, and use NaviQuest's cross-referencing capabilities to help you verify changes to the configuration.

See "Appendix E. Using NaviQuest" on page 321 for more information on DFSMS NaviQuest.

Displaying SMS and OAM Information

You can use the DISPLAY SMS operator command to determine the status of SMS, storage groups, DASD volumes, tape volumes, tape libraries, OAM, OSMC, optical libraries, optical volumes and optical drives.

- VARY SMS—for varying optical drives, optical libraries, or tape libraries online or offline
- DISPLAY SMS—for determining the status of an object or object backup storage group, an optical volume, optical library, optical drive, tape volume, tape library, or tape storage group; as well as for determining OAM, OSMC, OSMC task processing, or OAM XCF status

You can also enter the LISTSYS and LISTVOL line operators on the ISMF Storage Group List panels to get information about the currently active configuration, storage groups and volumes.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for specific information on using the OAM storage management component commands.

Displaying Information about the Active Configuration

To display information about the currently active configuration, issue the following command:

```
D SMS,ACTIVE or D SMS
```

where D is an abbreviation for DISPLAY. (The default for the DISPLAY command is to display the active configuration when no options are specified.) An example of the generated output appears in Figure 69 on page 180.

© Copyright IBM Corp. 1986, 2002

```
09.23.36          IGD002I 09:23:36 DISPLAY SMS 401

SCDS = SMS.SCDS1.SCDS
ACDS = SMS.ACDS1.ACDS
COMMDS = SMS.COMMDS1.COMMDS
DINTERVAL = 150
REVERIFY = NO
ACSDEFAULTS = NO
    SYSTEM     CONFIGURATION LEVEL    INTERVAL SECONDS
    SYSTEM01   1991/12/01 09:23:29           15
    SYSTEM02   ---------- --------          N/A
    SYSTEM03   ---------- --------          N/A
    SYSTEM04   ---------- --------          N/A
    SYSTEM05   ---------- --------          N/A
    SYSTEM06   ---------- --------          N/A
    SYSTEM07   ---------- --------          N/A
    SYSTEM08   ---------- --------          N/A
```

Figure 69. Displaying Information about the Active Configuration

The display shows the names of the current control data sets. The naming convention used here specifies a first level qualifier of SMS for all SMS control data sets. The second level qualifier identifies the type of SMS data set. The third level qualifier uniquely identifies the data set.

The DINTERVAL, REVERIFY ACSDEFAULTS, and OVRD_EXPDT fields contain the values given them when SMS was initialized. For an explanation of these values, see “Initializing SMS Through the IGDSMSxx Member” on page 14.

The last portion of the output shows configuration levels about the system from which you issued the DISPLAY command. The configuration level indicates the date and time the ACDS was last updated. This last portion also contains the synchronization time interval (the number of seconds that SMS allows before this system checks the COMMDS for news from other systems in the SMS complex) for each system. The synchronization time interval can be changed using the SET SMS command.

Displaying SMS TRACE Information

To display information about the SMS trace options in effect, issue the command:

```
D SMS,TRACE
```

where D is an abbreviation for DISPLAY. An example of the generated output appears in Figure 70 on page 181.

```
 14.57.24       IGD002I 14:57:24 DISPLAY SMS 879

 TRACE     = {ON|OFF}   SIZE =nnnK     TYPE {ERROR|ALL}
   JOBNAME ={jjj|*}    ASID ={asid|*}
   TRACING EVENTS:
MODULE ={ON|OFF}  SMSSJF ={ON|OFF}  SMSSI  ={ON|OFF}  ACSINT ={ON|OFF}
OPCMD  ={ON|OFF}  CONFC  ={ON|OFF}  CDSC   ={ON|OFF}  CONFS  ={ON|OFF}
MSG    ={ON|OFF}  ERR    ={ON|OFF}  CONFR  ={ON|OFF}  CONFA  ={ON|OFF}
ACSPRO ={ON|OFF}  IDAX   ={ON|OFF}  DISP   ={ON|OFF}  CATG   ={ON|OFF}
VOLREF ={ON|OFF}  SCHEDP ={ON|OFF}  SCHEDS ={ON|OFF}  VTOCL  ={ON|OFF}
VTOCD  ={ON|OFF}  VTOCR  ={ON|OFF}  VTOCC  ={ON|OFF}  VTOCA  ={ON|OFF}
RCD    ={ON|OFF}  DCF    ={ON|OFF}  DPN    ={ON|OFF}  TVR    ={ON|OFF}
DSTACK ={ON|OFF}
```

Figure 70. Displaying Trace Information

The display shows the status of the SMS trace option, the size of the SMS trace table, and the type of SMS trace entries. ERR means only error type trace entries are traced. ALL indicates all types of trace entries are traced. This section also includes the JOBNAME which indicates the tracing scope in relation to jobs being run. ASID indicates the tracing scope in relation to address spaces. ASID means tracing is limited to a particular address space and * indicates tracing is performed for all address spaces.

The last section, TRACING EVENTS, indicates which SMS events are selected for tracing.

You can use this tracing display if you have been requested to collect information by the IBM Support Center for diagnosing problems. See *z/OS DFSMSdfp Diagnosis Reference* for additional information.

Displaying Storage Group Status Using the DISPLAY SMS Command

You can use the STOGRP parameter to display the status of storage groups including tape and object storage groups. Figure 71 shows the syntax of the DISPLAY SMS,STORGRP command. See *z/OS MVS System Commands* for the complete syntax of the DISPLAY SMS command.

```
{DISPLAY} SMS{,STORGRP(STORGRP)} [LISTVOL|DETAIL] [,L={a}]
{D}          {,SG      (ALL)}                    [   {cc}]
                                                  [   {cca}]
                                                  [   {name}]
                                                  [   {name-a}]
```

Figure 71. DISPLAY SMS,STORGRP Command Syntax

STORGRP({storgrp|ALL})

If STORGRP(storgrp) is specified, the system displays the status of one storage group for each z/OS system connected to that storage group. If STORGRP(ALL) is specified, the system displays the status of all the storage groups defined in the SMS configuration. You can abbreviate STORGRP as SG.

LISTVOL
displays the status and volume serial numbers of all the volumes in the storage group. The LISTVOL parameter is ignored for tape, object, and object backup storage groups. This parameter is mutually exclusive with the DETAIL parameter.

DETAIL
displays detail status and is only valid for tape, object, and object backup storage groups. This parameter is mutually exclusive with the LISTVOL parameter and overrides the LISTVOL parameter if the storage group is tape or object related.

L=a|cc|cca|name|name-a
specifies where the results of the inquiry are to be displayed: the display area (a), the console (cc), or both (cca). The name parameter is routed to the console referred to by ″name″ and the name-a parameter is routed to the console referred to by ″name″ and the screen referred to by ″a″. The name parameter can be an alphanumeric character string.

To display information about the status of a storage group, issue the following command:

```
D SMS,STORGRP(storgrp)
```

For `storgrp` you specify the name of a single storage group or you specify `ALL` to display the status of all storage groups. An example of the generated output for the single storage group SG1 appears in Figure 72.

```
15.12.25          d sms,storgrp(slp01)
15.12.25          IGD0021 15:12:25 DISPLAY SMS 900
STORGRP TYPE              SYSTEM= 1 2 3 4 5 6 7 8
SG1     {DUMMY|POOL|TAPE|VIO}     + . . . . . . .
**************************** 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 = sysname1    SYSTEM 2 = sysname2    SYSTEM 3 = sysname3
SYSTEM 4 = sysname4    SYSTEM 5 = sysname5    SYSTEM 6 = sysname6
SYSTEM 7 = sysname7    SYSTEM 8 = sysname8    SYSTEM 9 = sysname9
SYSTEM 10= sysname10   SYSTEM 11= sysname11   SYSTEM 12= sysname12
SYSTEM 13= sysname13   SYSTEM 14= sysname14   SYSTEM 15= sysname15
SYSTEM 16= sysname16   SYSTEM 17= sysname17   SYSTEM 18= sysname18
SYSTEM 19= sysname19   SYSTEM 20= sysname20   SYSTEM 21= sysname21
SYSTEM 22= sysname22   SYSTEM 23= sysname23   SYSTEM 24= sysname24
SYSTEM 25= sysname25   SYSTEM 26= sysname26   SYSTEM 27= sysname27
SYSTEM 28= sysname28   SYSTEM 29= sysname29   SYSTEM 30= sysname31
SYSTEM 31= sysname31   SYSTEM 32= sysname32
```

Figure 72. Displaying Storage Group Status Information

If you have more than 16 systems in your SMS complex, a second message follows, and then the legend follows that.

Displaying the Status of Volumes in the Storage Group

Specifying the optional LISTVOL parameter provides the status of the volumes associated with the storage group. If you have volume 123456 in storage group SG1, the following command generates output similar to that shown in Figure 73.

```
D SMS,STORGRP(SG1),LISTVOL
```

```
15.12.25          d sms,storgrp(slp01)
15.12.25          IGD0021 15:12:25 DISPLAY SMS 900
STORGRP TYPE              SYSTEM= 1 2 3 4 5 6 7 8
SG1     {DUMMY|POOL|TAPE|VIO}     + . . . . . . .
VOLUME   UNIT    SYSTEM= 1 2 3 4 5 6 7 8 STORGRP NAME
123456                   + . . . . . . .   SG1
***************************** 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 = sysname1    SYSTEM 2 = sysname2    SYSTEM 3 = sysname3
SYSTEM 4 = sysname4    SYSTEM 5 = sysname5    SYSTEM 6 = sysname6
SYSTEM 7 = sysname7    SYSTEM 8 = sysname8    SYSTEM 9 = sysname9
SYSTEM 10= sysname10   SYSTEM 11= sysname11   SYSTEM 12= sysname12
SYSTEM 13= sysname13   SYSTEM 14= sysname14   SYSTEM 15= sysname15
SYSTEM 16= sysname16   SYSTEM 17= sysname17   SYSTEM 18= sysname18
SYSTEM 19= sysname19   SYSTEM 20= sysname20   SYSTEM 21= sysname21
SYSTEM 22= sysname22   SYSTEM 23= sysname23   SYSTEM 24= sysname24
SYSTEM 25= sysname25   SYSTEM 26= sysname26   SYSTEM 27= sysname27
SYSTEM 28= sysname28   SYSTEM 29= sysname29   SYSTEM 30= sysname31
SYSTEM 31= sysname31   SYSTEM 32= sysname32
```

Figure 73. Displaying Storage Group Volume Status Information

Displaying Storage Group Detail Status

If you specify the keywords DETAIL and STORGRP with a single storage group name, you get detailed information from OAM about the requested object, object backup, or tape storage group. If you specify DETAIL with ALL for the storage group name, you get detailed information from OAM about all object, object backup and tape storage groups.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for sample commands and outputs.

Displaying OAM Status in a Parallel Sysplex

Specifying the OAMXCF parameter displays the status of this instance of OAM within a Parallel Sysplex. You get the status of each member within the specific OAMplex, as well as the number of transactions waiting for a response from other instances of OAM in the Parallel Sysplex.

```
D SMS,OAMXCF
```

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information on object support.

Displaying the Caching Statistics

If you have an IBM 3990 Storage Control with cache with at least one SMS volume attached, you can display cache statistics, hit ratio, and DASD Fast Write bypasses, by issuing the following command:

```
D SMS,CACHE
```

An example of the generated output appears below.

```
IGD002I 09:28:38 DISPLAY SMS 411
  SSID  DEVS  READ  WRITE  HIT RATIO  FW BYPASSES
  0002   32    80    60        68%    62
  0004   14    92   100        76%    30
  000E   20    84   100        84%    44
  0011   08   100   100        92%    16
  001C   08   100   100        86%    24
**************************** LEGEND ****************************
SSID        THE SUBSYSTEM IDENTIFIER OF THE IBM 3990 STORAGE CONTROL
            MODEL 3 TO WHICH THE INFORMATION APPLIES
DEVS        THE NUMBER OF SMS-MANAGED DEVICES ATTACHED TO THE IBM 3990
            STORAGE CONTROL MODEL 3 WITH THIS SSID
READ        THE PERCENTAGE OF CYLINDERS FOR WHICH DYNAMIC CACHING IS
            PERMITTED
WRITE       THE PERCENTAGE OF CYLINDERS FOR WHICH DYNAMIC DASD FAST
            WRITE IS PERMITTED
HIT RATIO   THE READ HIT RATIO THAT IS ACHIEVED BY THE IBM 3990 STORAGE
            CONTROL MODEL 3 WITH THIS SSID
FW BYPASSES THE NUMBER DASD FAST WRITES PER MINUTE THAT ARE EXECUTED
            DIRECTLY WITH DASD BECAUSE OF MVS OVER-COMMITMENT
```

Figure 74. Displaying Cache and DASD Fast Write Information

Exception: The read and write percentages are displayed as N/A if the subsystem has the extended platform available. Since the enhanced dynamic cache management algorithm is used, the percentages are no longer valid.

Displaying Storage Group Status Using ISMF

You can display a storage group's status in each system in the SMS complex by using the LISTSYS line operator in ISMF. The storage group must be a pool type, and it must be part of the active configuration. Go to the Storage Group Application Selection panel, shown in Figure 11 on page 45, from the ISMF Primary Option Menu. Then create a list of storage groups. For example, specifying the following information on the Storage Group Application Selection panel creates a list of storage groups whose names begin with SGNAME:

```
CDS NAME             ===> 'ACTIVE'
STORAGE GROUP NAME   ===> SGNAME*
SELECT OPTION        ===> 1
```

Press the Enter key to display the Storage Group List panel, and enter LISTSYS in the Line Operator column next to the desired pool type storage group. Press Enter to display the Storage Group System Status Display panel shown in Figure 75 on page 185.

Requirement: If a volume has been initialized with LABEL as an SMS device, the storage group must be defined in the active configuration in order for space information to be displayed when the LISTSYS line operator is issued.

```
  Control  Utilities  Scroll  Help
--------------------------------------------------------------------------------
                        SMS STORAGE GROUP STATUS DISPLAY            Page 1 of 2
Command ===>

CDS Name . . . . . : ACTIVE
Storage Group Name : SGNAME01           * SYS GROUP = sysplex minus systems in
Total Space for Storage Group             sysplex explicitly defined in SCDS
 Mb-total: -------
 Mb-free : -------   Available Group                        Available Group
 % Free  :  --    Space For Allocation                    Space For Allocation
System/Sys SMS SG Mb-     Mb-     %     System/Sys SMS SG Mb-     Mb-     %
Group Name Status Total   Free    Free  Group Name Status Total   Free    Free
---------- ------ ------- ------- ----  ---------- ------ ------- ------- ----
*SYSPX1     ENABLE ------- -------   --  SYSTEM2    NOTCON ------- -------   --
*SYSPX2     NOTCON ------- -------   --  SYSTEM4    NOTCON ------- -------   --
 SYSTEM5    NOTCON ------- -------   --  SYSTEM6    NOTCON ------- -------   --
 SYSTEM7    NOTCON ------- -------   --  SYSTEM8    NOTCON ------- -------   --




Use HELP Command for Help; Use DOWN Command to View Next Panel; END to Exit.
```

Figure 75. Displaying Storage Group Status for Each System/System Group

The Total Space for Storage Group value is the total capacity of all the online volumes belonging to storage group SGNAME01, of all the volumes that have a status of ENABLE.

MB-TOTAL Total number of MB belonging to the storage group

MB-FREE Total number of free MB belonging to the storage group

% FREE Percentage of free storage group space

Requirement: Space information exists for a volume only if the volume has at least one SMS-managed data set allocated on it.

The Group Space Available for Allocation value represents the sum of all volume space that is currently available to each system online to MVS. The fields contain values only if the SMS SG Status value is ENABLE.

MB-TOTAL Total available space in MB

MB-FREE Total available space in MB that is free

% FREE Percentage of total available space that is free

Displaying Volume Status Using ISMF

You can display a volume's status by going to the Storage Group Application Selection panel, shown in Figure 11 on page 45 from the ISMF Primary Option Menu. Select option 4, VOLUME, and the Storage Group Selection panel is displayed. After entering the single volume or range of volumes you wish to display, choose option 1 (Display) to access the SMS Volume Status Display panel shown in Figure 76 on page 186.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                           SMS VOLUME STATUS DISPLAY              Page 1 of 2
Command ===>

CDS Name . . . . . . . . : ACTIVE
Storage Group Name . . : S1P01
Volume Serial Number . : 1P0101
Physical Volume Status : CONVERTED        * SYS GROUP = sysplex minus systems
               Mb-free : -------            in the sysplex explicitly defined
               % Free  : --                 in the SCDS

  System/Sys SMS Vol MVS Vol  SMS SG  |  System/Sys SMS Vol MVS Vol  SMS SG
  Group Name Status  Status   Status  |  Group Name Status  Status   Status
  ---------- ------- -------- ------  |  ---------- ------- -------- ------
  SYSTEM1    ENABLE  ONLINE   ENABLE  |  SYSTEM2    ENABLE  -------- NOTCON
  SYSTEM3    ENABLE  -------- NOTCON  |  SYSTEM4    NOTCON           NOTCON
  SYSTEM5    NOTCON           NOTCON  |  SYSTEM6    NOTCON           NOTCON
  SYSTEM7    NOTCON           NOTCON  |  SYSTEM8    NOTCON           NOTCON
                                      |
                                      |
                                      |
Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 76. Displaying the SMS Volume Status

The Physical Volume Status field shows the current SMS status of, and the amount of free space on, the volume. The volume status can be:

INITIAL The volume has been defined to a storage group, but all the data sets are not yet under SMS control.

CONVERTED The volume is fully converted to SMS.

NONSMS The volume is not under SMS control.

UNKNOWN The status of the volume is not known.

The Mb-free value is the total number of free megabytes belonging to the volume.

The % Free value is the percentage of free volume space.

Requirement: If you do not specify 'ACTIVE' in the CDS Name field, then the Physical Volume Status, Mb-free, % Free and MVS Vol Status fields display dashes. The space information only exists for a volume if the volume has at least one SMS-managed data set allocated on it.

The SMS Vol Status field shows the relationship between the volume and SMS. The six possible relationships are:

DISALL SMS does not permit data sets on this volume to be accessed.

DISNEW SMS does not permit the allocation of new data sets on this volume.

ENABLE SMS permits access to data sets on this volume.

NOTCON SMS does not attempt to access this volume.

QUIALL SMS does not schedule any more jobs that access data sets on this volume.

QUINEW SMS does not schedule any jobs that create new data sets on this volume.

The MVS Vol Status column shows relationships between MVS and the volume whose status is being displayed. Possible relationships are:

BOXED A status encountered when an application cannot disconnect from a malfunctioning volume. BOXED means that MVS is simulating I/O errors in response to the application's I/O request.

NOTREADY The volume cannot send nor receive I/O now.

OFFLINE I/O is not possible because the storage management subsystem cannot find the address where the volume's VOLSER is mounted.

ONLINE The volume is physically connected to MVS; I/O can proceed.

PENDOFF (Pending Offline) MVS varies the volume offline as soon as persons currently accessing data sets on it have disconnected from it.

Displaying Volumes Using the DISPLAY SMS Command

To display the status of specific DASD, tape or optical disk volumes issue the following command:

```
D SMS,VOLUME(serial-number)
```

If the volume serial number is of a DASD volume in a pool storage group, you get the generated output on the Operator Display panel shown in Figure 77. In this example, the volume serial number is 123456.

```
15.38.00          IGD002I 15:38:00 DISPLAY SMS 918
VOLUME   UNIT    SYSTEM= 1 2 3 4 5 6 7 8 STORGRP NAME
123456   123             + . . . . . . .   SG1
**************************** 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 = sysname1    SYSTEM 2 = sysname2     SYSTEM 3 = sysname3
SYSTEM 4 = sysname4    SYSTEM 5 = sysname5     SYSTEM 6 = sysname6
SYSTEM 7 = sysname7    SYSTEM 8 = sysname8     SYSTEM 9 = sysname8
SYSTEM 10= sysname10   SYSTEM 11= sysname11    SYSTEM 12= sysname12
SYSTEM 13= sysname13   SYSTEM 14= sysname14    SYSTEM 15= sysname15
SYSTEM 16= sysname16   SYSTEM 17= sysname17    SYSTEM 18= sysname18
SYSTEM 19= sysname19   SYSTEM 20= sysname20    SYSTEM 21= sysname21
SYSTEM 22= sysname22   SYSTEM 23= sysname23    SYSTEM 24= sysname24
SYSTEM 25= sysname25   SYSTEM 26= sysname26    SYSTEM 27= sysname27
SYSTEM 28= sysname28   SYSTEM 29= sysname29    SYSTEM 30= sysname30
SYSTEM 31= sysname31   SYSTEM 32= sysname32
```

Figure 77. Displaying Status Information for Individual DASD Volumes

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for information about displaying the status of optical and tape volumes.

Displaying the Device Status

You can use the DEVSERV command to request and display basic status information on a device, a group of devices, or storage control units.

See *z/OS MVS System Commands* for a detailed description of the syntax and parameters of the DEVSERV command.

DEVSERV SMS

You can use the DEVSERV SMS or DEVSERV S command to display the volume and storage group status of *nn* devices that SMS manages, starting with the specified device number.

This is the syntax for the DEVSERV SMS command:

```
{DEVSERV} {SMS},ddd[,nn][,ONLINE][,L={a}]
{DS}      {S}           [,ON]    [   {cc}]
                                 [   {cca}]
                                 [   {name}]
                                 [   {name-a}]
```

Where:

ddd specifies the device number, in hexadecimal, for which the system is to display information.

nn specifies the decimal number (1 - 32) of devices for which the system is to display the information, in ascending order beginning with the device you specify. If you do not code *nn*, the system displays information about the one device you specify.

ONLINE/ON
displays information about only those specified devices that are online. If you do not specify ONLINE or OFFLINE, the system displays information about both online and offline devices.

OFFLINE/OFF
displays information about only those specified devices that are offline.

L=a|cc|cca|name|name-a
specifies where the results of the inquiry are to be displayed: the display area (a), the console (cc), or both (cca). The name parameter is routed to the console referred to by ″name″ and the name-a parameter is routed to the console referred to by ″name″ and the screen referred to by ″a″.

Figure 78 shows the output of the DEVSERV S,430 command for the target device 430.

```
15.44.26          IGD001I 15:44:26 DEVSERV SMS 921
 UNIT,DTYPE   ,M,VOLSER,VOLSTAT    STORGRP,SGSTAT
 430 ,3380    ,O,XP0101,ENABLED    SXP01,QUIESCED

******************************LEGEND******************************
A = ALLOCATED                     F = OFFLINE
M = MOUNT PENDING                 N = NOT ALLOCATED
O = ONLINE                        P = PENDING OFFLINE
```

Figure 78. DEVSERV S,430 Output

DEVSERV QPAVS

QPAVS is the new parameter of the DEVSERV command to support the parallel access volume (PAV) capability of the IBM Enterprise Storage Server (ESS). You can use the DEVSERV QPAVS command to:

- Describe how a logical subsystem configuration is defined to z/OS.
- Highlight the inconsistencies, if any, between the IODF and the LSS configuration.
- Display unbound alias device types with the UCB parameter and, if necessary, ″unbox″ a boxed alias device with the UNBOX parameter.
- Show information on both a PAV-base address and its PAV-aliases by specifying the VOLUME parameter.
- Display information on devices.

Based on your service needs, you can issue the DEVSERV QPAVS command in the forms as shown in Figure 79:

```
{DEVSERV} {QPAVS},dddd[,nn]
{DS}                  [,tttt]
                 SSID[=ssss]
```

Figure 79. DEVSERV SMS Command Syntax

Where:

QPAVS	is a required positional keyword.
dddd	is a 3 or 4-hex-digit device number.
nn	is a decimal number from 1 - 256 with 1 as the default.
tttt	can be UCB, VOLUME, or UNBOX.
SSID	is the logical volume ID of the storage subsystem.
ssss	is the specified SSID value.

Figure 80 on page 190 shows the display content when you issue the DEVSERV QPAVS command with parameters other than UCB. When you specify the UCB parameter, you receive additional UCB information. This additional information is formatted the same way as the output resulting from the existing QDASD parameter.

```
-----------------------------------------------------
---          DS QP  Display Content               ---
-----------------------------------------------------
IEE459I  (time)   DEVSERV QPAVS
    Host                              Subsystem
Configuration                       Configuration
-------------                     --------------------
UNIT                                    UNIT    UA
NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
---- --  ----        ------     ----  ----    --------
dddd aa  BASE        INV-ALIAS  nnnn   uu     BASE
         ALIAS-bbbb  NOT-BASE                 ALIAS-bb
         NON-PAV     NOT-ALIAS                NC
                     NOT-NPAV

UNLISTED DEVICES AND REASON CODES  X
   nnnn(rc) nnnn(rc) nnnn(rc) ...
```

Figure 80. DEVSERV QPAVS Display Content

X Below are the reason codes for unlisted devices:

```
(01) - DEVICE NOT CONFIGURED, UCB NOT FOUND
(02) - UCB NOT CONNECTED
(03) - DEVICE UNAVAILABLE, SCP ROUTINE IN CONTROL
(04) - SUBCHANNEL ERROR
(05) - DEVICE BOXED
(06) - UCB NOT A DASD
(07) - DEVICE I/O ERROR
(08) - DEVICE IS NOT A DASD
(09) - DSE-1 CCW BUILD FAILED
(0A) - DEVICE IS AN UNBOUND PAV-ALIAS
```

The following are examples of DEVSERV QPAVS output.

Figure 81 shows the PAV status for the device at the starting address D123 and the next 2 addresses.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
    Host                              Subsystem
Configuration                       Configuration
-------------                     --------------------
UNIT                                    UNIT    UA
NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
---- --  ----        ------     ----  ----   -------
D123 23  NON-PAV                0101   23     BASE
D124 24  NON-PAV                0101   24     BASE
D125 25  NON-PAV                0101   25     BASE
****      3 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 81. DS QP,D123,3 Output

Figure 82 on page 191 shows the PAV status for the alias device at the address D2FF, its base, and other associated alias devices of the logical volume.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
    Host                              Subsystem
Configuration                       Configuration
-------------                     --------------------
UNIT                                    UNIT    UA
NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
---- --  ----        ------     ----  ----    --------
D222 22  BASE                   0102   22     BASE
D2FE FE  ALIAS-D222             0102   FE     ALIAS-22
D2FF FF  ALIAS-D222             0102   FF     ALIAS-22
****      3 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 82. DS QP,D2FF,VOLUME Output

Figure 83 shows the PAV status for the base device at the address D222 and its alias volumes.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
    Host                              Subsystem
Configuration                       Configuration
-------------                     --------------------
UNIT                                    UNIT    UA
NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
---- --  ----        ------     ----  ----    --------
D222 22  BASE                   0102   22     BASE
D2FE FE  ALIAS-D222             0102   FE     ALIAS-22
D2FF FF  ALIAS-D222             0102   FF     ALIAS-22
****      3 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 83. DS QP,D222,VOLUME Output

Figure 84 shows that you can ″unbox″ a boxed alias device at the address D6FF.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
 THE DEVSERV QPAV UNBOX COMMAND HAS BEEN EXECUTED SUCCESSFULLY 1
```

```
IEE459I 08.20.32 DEVSERV QPAVS 591
 THE DEVSERV QPAV UNBOX COMMAND HAS BEEN EXECUTED WITH
   RETURN CODE = xx, REASON CODE = yy 2
```

```
IEE459I 10.04.23 DEVSERV QPAVS 699
 D6FF  IS NOT AN UNBOUND PAV-ALIAS DEVICE. 3
   THE DEVSERV QPAV UNBOX COMMAND IS NOT EXECUTED
```

Figure 84. DS QP,D6FF,UNBOX Output

1 You receive this message if the return code from DEVSERV QPAVS UNBOX is zero.

2 You receive this message if the return code from DEVSERV QPAVS UNBOX is nonzero.

3 You may also receive this message with the explanation: ″nnnn-IS NOT IN BOX STATE.″

Figure 85 describes the status of an unbound alias and the display of its associated UCB control blocks.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
     Host                              Subsystem
 Configuration                      Configuration
 -------------                    --------------------
 UNIT                                   UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----    ---------
   UCB AT V01ED2AA8
 0088FF04D5EF0000 0000000008E4C3C2 3010200E00ED2A81 0000000000000000
 0000000000100000 0000000000000000 0000000000000000
   UCB PREFIX AT V01FF4208
 000C804000000000 000000000001004E 289C1253C00080C0 14F4FFFFFFFFFFFF
 0148000000000001
   UCB COMMON EXTENSION AT V01ED2A80
 00000900182A0000 01FF420800000000 0000000000FCD3D8 01ECA90000000000
**** UNLISTED DEVICE(S) AND REASON CODES:
  D5EF(0A)
**** (0A) - DEVICE IS AN UNBOUND PAV-ALIAS
```

Figure 85. DS QP,D5EF,UCB Output

Figure 86 shows the PAV status of the devices that have the same SSID value.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
     Host                              Subsystem
 Configuration                      Configuration
 -------------                    --------------------
 UNIT                                   UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----    --------
 D400 00  NON-PAV                1401   00     BASE
 ....
 D422 22  BASE                   1401   22     BASE
 ....
 D4FF FF  ALIAS-D422             1401   FF     ALIAS-22
****    256 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 86. DS QP,SSID=1401 Output

Figure 87 on page 193 shows the status of NOT-BASE when the address D345 is defined as a PAV-base in the HCD, but not in the ESS logical subsystem.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
     Host                              Subsystem
 Configuration                       Configuration
 -------------                    --------------------
 UNIT                                   UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----    --------
 D345 45  BASE        NOT-BASE   0103   45     ALIAS-00
****      1 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 87. DS QP,D345 Output

Figure 88 shows the status of NOT-ALIAS when the address D621 is defined as a PAV-alias in the HCD, but not in the ESS logical subsystem.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
     Host                              Subsystem
 Configuration                       Configuration
 -------------                    --------------------
 UNIT                                   UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----   ---------
 D621 21  ALIAS-D600  NOT-ALIAS  0106   21     BASE
****      1 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 88. DS QP,D621 Output

Figure 89 shows the status of INV-ALIAS when the alias address D6F4 for the base volume in the HCD does not match its base address in the ESS logical subsystem.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
     Host                              Subsystem
 Configuration                       Configuration
 -------------                    --------------------
 UNIT                                   UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----    --------
 D6F4 F4  ALIAS-D600  INV-ALIAS  0106   F4     ALIAS-06
****      1 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 89. DS QP,D6F4 Output

Figure 90 on page 194 shows the status of NOT-NPAV for the device at the address F60. The device is defined as a NON-PAV device in the HCD, but is given an alias in the ESS logical subsystem.

```
IEE459I 13.20.12 DEVSERV QPAVS 368
     Host                                Subsystem
 Configuration                        Configuration
 -------------                      --------------------
 UNIT                                     UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ------      ------     ----  ----    --------
 0F60 F0  NON-PAV     NOT-NPAV   0101   F0     ALIAS-03
```

Figure 90. DS QP,F60 Output

DEVSERV PATHS

You can issue the DEVSERV PATHS command to display channel path IDs and status, the logical mode of devices, the number of data sets allocated on volumes, and volume serial labels. The displayed information can help you solve hardware or configuration problems.

Figure 91 shows the output of the DS P,A000 command.

```
DS P,A000
IEE459I 07.51.43 DEVSERV PATHS 454
UNIT DTYPE  M CNT VOLSER  CHPID=PATH STATUS
     RTYPE   SSID CFW TC   DFW   PIN  DC-STATE CCA  DDC   ALT  CU-TYPE
A000,33903 ,O,000,SKA000,4A=+ 5A=< 4B=< 5B=<
     2105    3201  Y  YS.  YY.    N   SIMPLEX   00   00        2105
```

Figure 91. DS P,A000 Output

DEVSERV QDASD

You can use the DEVSERV QDASD command to display diagnostic information about the status of direct access storage devices and storage control units. You can also use it to validate MVS storage resident control blocks for extended function status with the data acquired directly from the storage subsystem.

Figure 92 shows the output of the DS QD,D407,RDC command.

```
DS QD,D407,RDC IEE4591 14.09.06 DEVSERV QDASD 374
UNIT VOLSER SCUTYPE DEVTYPE   CYL  SSID SCU-SERIAL DEV-SERIAL EF-CHK
D407 339331 2105F20    2105 32760  8105 0113-FCA81 0113-FCA81 **OK**
  READ DEVICE CHARACTERISTIC
2105E833900C5E80 FFF720327FF8000F E000E5A205940222 1309067400000000
0000000000000000 32321B02DFEE0001 0677080F007F4A00 0035000000000000
****      1 DEVICE(S) MET THE SELECTION CRITERIA
****      0 DEVICE(S) FAILED EXTENDED FUNCTION CHECKING
```

Figure 92. DS QD,D407,RDC Output

Listing SMS Classes, Aggregate Groups, Storage Groups, and Libraries Using ISMF

You can list SMS classes, aggregate groups, storage groups, optical libraries, optical drives, saved lists, DASD volumes, optical volumes, tape libraries, tape volumes, and data sets on application panels. You can also generate listings by selecting the LIST option on application panels or by typing the LIST line operator. "Appendix A. Quick Reference to ISMF Commands and Line Operators" on page 295 summarizes all the ISMF commands and line operators, describes them, and lists the applications from which you can issue them.

ISMF supports the following screen sizes for various lists:

24 x 80
27 x 132
32 x 80
43 x 80
31 x 160

The examples shown in this book are for displays with a screen size of 24 x 80.

Note: 31 x 160 is half of a 3290 screen. Only half of the 3290 screen is used.

Listing with View and Sort

When you specify the list option on the Data Class, Management Class, Storage Class, Storage Group, Aggregate Group, Library, or Drive Application Selection panels, the values you specify in the Respecify View Criteria field and the Respecify Sort Criteria field determine characteristics of the list that ISMF displays.

If you specify yes, Y, in the Respecify View Criteria field of the Application Selection panel, you see the View Entry panel for that application. You can specify which data columns appear in the list and the order in which they appear, and you can save your specifications to use again. If you specify no, N, in the Respecify View Criteria field, ISMF uses the last used values for data column selection and order. The initial value is all data columns in their default order.

If you specify yes, Y, in the Respecify Sort Criteria field of the Application Selection panel, you see the Sort Entry panel for that application. You can specify by which data columns entries in the list are sorted. You can only specify data columns that are already specified as view criteria. If you specify no, N, in the Respecify Sort Criteria field, ISMF uses the default sorting order.

"Listing Data Classes" gives an example of how the View and Sort panels work together. See *z/OS DFSMS: Using the Interactive Storage Management Facility* for detailed information about View and Sort.

Listing Data Classes

You can use View and Sort to customize the way you see a list. For example, if you want to see a list of data classes with their Data Set Name Type, Logical Record Length, and Record Organization data columns displayed, and you want the entries to appear sorted by data set name type in ascending order, you can follow the panels in this section.

Choose the LIST option from the Data Class Application Selection panel, shown in Figure 36 on page 112, and select the viewing order by specifying Y, yes, in the Respecify View Criteria field. Select the sorting order by specifying Y, yes, in the

Respecify Sort Criteria field. ISMF displays the Data Class View Entry panel. If you want to see Data Set Name Type, Logical Record Length, and Record Organization, type their corresponding tags, separated by blanks, in the field called Specify Tags in Sequence Desired, as shown in Figure 93. Your selections are displayed in the order that you typed them.

Line Operator and Data Class Name always appear as the first and second data columns in a list but, because you cannot select them from this panel, they do not have corresponding tags. You cannot specify them on the this panel as selected tags.

You can choose option 2, Save, to save your specifications for later use.

Figure 93 shows page 1 of the Data Class View Entry Panel.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DGTDVW41                DATA CLASS VIEW ENTRY PANEL              Page 1 of 3
Command ===>

If desired, Select option . .           (1 - SELECT, 2 - SAVE, 3 - DELETE)
                View Name . .           (1 to 8 characters, * to Delete all,
                                          or blank for List of Saved Views)
Specify tags in Sequence Desired:
===> *

       Line Operator                 (26)  Data Set Name Type
       Data Class Name               (43)  Dynamic Volume Count
 (15)  Additional Volume Amt         (27)  Extended Addressability
  (8)  Avgrec                        (41)  FRlog
  (9)  Avg Value                     (16)  Imbed
 (34)  BWO                           (32)  Initial Load
 (40)  Block Size Limit               (6)  Key Length
 (28)  Compaction                     (7)  Key Offset
 (18)  Control Interval Size data    (24)  Last Date Modified

Use ENTER to Perform Selection or Display List in VIEW Sequence;
Use DOWN for next Panel; Use HELP Command for Help; Use END Command to Exit.
```

Figure 93. Specifying Data Class View Criteria, Page 1 of 3

Figure 94 on page 197 shows page 2 of the Data Class View Entry Panel.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
DGTDVW42                DATA CLASS VIEW ENTRY PANEL              Page 2 of 3
Command ===>

If desired, Select option . .           (1 - SELECT, 2 - SAVE, 3 - DELETE)
                View Name . .           (1 to 8 characters, * to Delete all,
                                          or blank for List of Saved Views)
Specify tags in Sequence Desired:
===> *

 (23)  Last Modified Userid                  (4)  Record Format
 (25)  Last Time Modified                    (3)  Record Organization
 (35)  Log                                  (30) Recording Technology
 (5)   Logical Record Length                (38) Reduce Space Up To (%)
 (36)  Logstream Id                         (17) Replicate
 (39)  Media Type                           (13) Retpd or Expdt
 (19)  % Freespace Control Area             (42) RLS CF Cache Value
 (20)  % Freespace Control Interval         (31) Reuse
 (39)  Record Access Bias                   (21) Shareoptions Crossregion

Use ENTER to Perform Selection or Display List in VIEW Sequence;
Use UP For previous Panel; Use HELP Command for Help; Use END Command to Exit.
```

Figure 94. Specifying Data Class View Criteria, Page 2 of 3

Figure 95 shows page 3 of the Data Class View Entry Panel.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
DGTDVW42                DATA CLASS VIEW ENTRY PANEL              Page 3 of 3
Command ===>

If desired, Select option . .           (1 - SELECT, 2 - SAVE, 3 - DELETE)
                View Name . .           (1 to 8 characters, * to Delete all,
                                          or blank for List of Saved Views)
Specify tags in Sequence Desired:
===> *

 (22)  Shareoptions Crosssystem                (11)  Space Secondary
 (37)  Space Constraint Relief                 (33)  Spanned / Nonspanned
 (12)  Space Directory                         (14)  Volume Count
 (10)  Space Primary

Use ENTER to Perform Selection or Display List in VIEW Sequence;
Use UP For previous Panel; Use HELP Command for Help; Use END Command to Exit.
```

Figure 95. Specifying Data Class View Criteria, Page 3 of 3

Next, ISMF displays the Data Class Sort Entry panels. Enter 25 as the Major Field Sequence and A as the Major Field Order as shown in Figure 96 on page 198 and Figure 97 on page 198 to sort in ascending order by Data Set Name Type. Tags for Line Operator and Data Class Name always appear on this panel. Notice that the only other tags displayed are the ones you specified in the Data Class View Entry panel.

Figure 96 on page 198 shows page 1 of the Data Class Sort Entry Panel.

```
  Panel  Utilities  Scroll  Help
--------------------------------------------------------------------------------
DGTDCDC3               DATA CLASS SORT ENTRY PANEL                Page 1 of 3
Command ===>

Specify one or more Attribute Numbers for Sort Sequence:
  Major Field  . . . 2    Minor Field 1 . . .       Minor Field 2 . . .

Specify A for Ascending or D for Descending Sort Order:
  Major Field  . . . A    Minor Field 1 . . .       Minor Field 2 . . .

   (1)  Line Operator                (26)  Data Set Name Type
   (2)  Data Class Name              (43)  Dynamic Volume Count
  (15)  Additional Volume Amt        (27)  Extended Addressability
   (8)  Avgrec                       (41)  FRlog
   (9)  Avg Value                    (16)  Imbed
  (34)  BWO                          (32)  Initial Load
  (40)  Block Size Limit              (6)  Key Length
  (28)  Compaction                    (7)  Key Offset
  (18)  Control Interval Size data   (24)  Last Date Modified

Use ENTER to Perform SORT; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 96. Specifying Data Class Sort Criteria, Page 1 of 3

Figure 97 shows page 2 of the Data Class Sort Entry Panel.

```
  Panel  Utilities  Scroll  Help
--------------------------------------------------------------------------------
DGTDCDC4               DATA CLASS SORT ENTRY PANEL                Page 2 of 3
Command ===>

Specify one or more Attribute Numbers for Sort Sequence:
  Major Field  . . . 2    Minor Field 1 . . .       Minor Field 2 . . .

Specify A for Ascending or D for Descending Sort Order:
  Major Field  . . . A    Minor Field 1 . . .       Minor Field 2 . . .

  (23)  Last Modified Userid                  (4)  Record Format
  (25)  Last Time Modified                    (3)  Record Organization
  (35)  Log                                (30)  Recording Technology
   (5)  Logical Record Length              (38)  Reduce Space Up to (%)
  (36)  Logstream Id                       (17)  Replicate
  (29)  Media Type                         (13)  Retpd or Expdt
  (19)  % Freespace Control Area           (42)  RLS CF Cache Value
  (20)  % Freespace Control Interval       (31)  Reuse
  (38)  Record Access Bias                 (21)  Shareoptions Crossregion

Use ENTER to Perform SORT; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 97. Specifying Data Class Sort Criteria, Page 2 of 3

Figure 98 on page 199 shows page 3 of the Data Class Sort Entry Panel.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
DGTDCDC4              DATA CLASS SORT ENTRY PANEL                Page 3 of 3
Command ===>

Specify one or more Attribute Numbers for Sort Sequence:
  Major Field  . . . 2    Minor Field 1 . . .       Minor Field 2 . . .

Specify A for Ascending or D for Descending Sort Order:
  Major Field  . . . A    Minor Field 1 . . .       Minor Field 2 . . .

 (22)  Shareoptions Crosssystem                      (11)  Space Secondary
 (37)  Space Constraint Relief                       (33)  Spanned / Nonspanned
 (12)  Space Directory                               (14)  Volume Count
 (10)  Space Primary

Use ENTER to Perform SORT; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 98. Specifying Data Class Sort Criteria, Page 3 of 3

On the basis of the view and sort criteria, ISMF displays a list of data classes like the one shown in Figure 99. On the listing, you can enter line operators such as COPY, DISPLAY, and HIDE against individual entries. Some line operators, such as ALTER and DELETE, cannot be used with the ACTIVE control data set.

```
  Panel  List  Utilities  Scroll  Help
-----------------------------------------------------------------------------------------
                                DATA CLASS LIST
Command ===>                                                 Scroll ===> CSR
                                                        Entries 1-9 of 16
                                                        Data Columns 38-43 of 43
CDS Name : Y421252.MYSCDS

Enter Line Operators below:

     LINE       DATACLAS REDUCE   REC ACC  BLOCK SIZE          RLS CF      DYNVOL
     OPERATOR   NAME     SPC UPT  BIAS     LIMIT        FRLOG  CACHE       COUNT
    ---(1)----  --(2)--- -(38)--  -(39)--  ---(40)---   -(41)- ---(42)---  -(43)-
                ABC1     --       USER     ---------    NONE   ALL            --
                DCA11    --       ------       163800   NONE   NONE           --
                DCA7     --       ------   ---------    NONE   UPDATESONLY    --
                DCC1     --       USER     ---------    NONE   ALL            --
                DCC7     --       ------   ---------    NONE   NONE           --
                DCT1     --       ------       458640   NONE   ALL             1
                DCY      --       USER     ---------    NONE   NONE           --
                DC1      --       ------   ---------    NONE   ALL            12
                DC2      --       ------   ---------    NONE   ALL            --
```

Figure 99. Displaying a List of Data Classes

You can scroll up, down, left, and right. However, ISMF always displays the LINE OPERATOR and DATACLAS NAME fields on the left side of the panel. You can also change the View and Sort criteria by issuing the View or Sort command on the command line.

Listing Storage Groups, Management Classes, Storage Classes, Aggregate Groups, Libraries and Drives

The viewing, sorting and saving procedures for lists of storage groups, storage classes, and libraries correspond to the example of the procedures for data classes.

Processing Management Classes

With the introduction of ABARS II the number of columns on the Management Class list was modified to include columns 31 through 38. Even though the number of attributes has increased, the Display, Define and Alter functions are simplified with the reorganization of the current and new management class attributes into four separate attribute groups: Space, Backup, Aggregate Backup, and Class Transition. When displaying, defining, or altering a management class, you can specify which set of attributes to process first.

Altering Data Set Associations

You can change the management class or storage class associated with a data set. The management class or storage class you choose for the data set must be in the ACDS. Enter the ALTER line operator against a data set entry on a Data Set List (option 1 in the ISMF Primary Option Menu).

Altering SMS Classes, Aggregate Groups, Storage Groups, Libraries, Volumes, and Drives

You can change definitions of SMS classes, storage groups, volumes, aggregate groups, optical libraries, optical drives, and tape libraries by selecting the alter option from the Application Selection panels. You can also change the definitions by entering the ALTER line operator against a list of constructs. This method is more efficient if you want to make multiple changes in one SCDS.

Prerequisite: You must be in storage administrator mode to be able to alter SMS classes. If you are in user mode, you are not authorized to alter SMS classes and libraries.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for information on altering optical libraries and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for information on altering tape libraries and drives.

Altering Storage Groups

As the needs of your installation change, you might need to modify storage groups. To modify a storage group, go to the Storage Group Application Selection panel, shown in Figure 11 on page 45, and specify the name of the SCDS containing the storage group you want to modify. Specify the storage group name and select option 3, ALTER. ISMF displays the Storage Group Alter panel.

You can also use the SETSMS operator command to change the SMS status of storage groups.

Restriction: Any SMS status that is changed using the SETSMS command is overridden by the MVS status if a new configuration is activated. See *z/OS MVS System Commands* for more information on the SETSMS operator command.

Redefining Volumes Among Storage Groups

If you have defined DASD volumes to belong to one storage group but now want them to belong to a different storage group, go to the Storage Group Application Selection panel. Specify the CDS name and storage group name, then select option 4, Volume. Press Enter to get the Storage Group Volume Selection panel shown in Figure 100.

```
  Panel  Utilities  Help
--------------------------------------------------------------------------------
                         STORAGE GROUP VOLUME SELECTION
Command ===>

CDS Name  . . . . . : SMS.SCDS1.SCDS
Storage Group Name  : PRIMARY
Storage Group Type  : POOL

Select One of the following Options:
  4  1. Display      - Display SMS Volume Statuses (Pool only)
     2. Define       - Add Volumes to Volume Serial Number List
     3. Alter        - Alter Volume Statuses   (Pool only)
     4. Delete       - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
      Prefix   From     To    Suffix  Hex
      ______  ______  ______  _____    _
 ===> SYS     001     077                   ('X' in HEX field allows
 ===> DFPIP1                                 FROM - TO range to include
 ===> SY      25      30      A        X      hex values A through F.)
 ===>         001     010     VOL
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 100. Moving a Volume from One Storage Group to Another

In the Specify a Single Volume (in Prefix), or Range of Volumes field, enter the serial numbers of the volumes you want to redefine, select option 4 to delete them from their current storage group, and press Enter. Then, return to the Storage Group Application Selection panel, enter the name of the new storage group to contain the volumes, and select option 4, Volume, which returns you to the Storage Group Volume Selection panel. Specify the volume serial numbers, select option 2 to define them to the storage group, and press Enter. You need to activate the updated configuration.

You are responsible for maintaining consistent physical connections and being aware of multivolume data sets. All of a multivolume data set or a VSAM sphere (the base cluster and its associated alternate indexes) must be in a single storage group. You must also maintain storage group, management class, and storage class consistency. If a data set has a management class with certain migration and backup attributes, and you redefine the volume containing that data set from one storage group to another, the new storage group might not be eligible for the same migration and backup processing.

You can also use the SETSMS operator command to change the SMS status of volumes. Note, however, that any SMS status that is changed in this way is overridden by the MVS status if a new configuration is activated.

Altering Management Classes

As the needs of your installation change, you might need to modify management classes. To modify a management class, go to the Management Class Application Selection panels, shown in Figure 23 on page 72 and Figure 24 on page 73, and specify the name of the SCDS containing the management class you want to change. Specify the management class name and select option 4, Alter. ISMF displays the Management Class Alter panel shown in Figure 101.

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

SCDS Name . . . . . . : SMS.ONE.TWO
Management Class Name : MCUSER01

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


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 101. Altering a Management Class

This panel contains the same attributes as the Management Class Define panel. The fields are primed with the information you entered when you defined the management class. To alter the management class, type over the old information with your new information on this panel and on the next three Management Class Alter panels. You can use the DOWN command to reach the next panels and use the END command to save the changes and return to the Management Class Application Selection panel. The changes take effect when you activate this updated configuration.

The # Versions field specifies the number of versions of data sets assigned this management class to be maintained.

The Retain Only Version and Retain Extra Versions fields specify the time period for which the copy is to be retained. The unit of measure for the time period is specified in the accompanying Unit field. This field cannot be blank if the Unit field is specified.

The Copy Serialization field specifies whether aggregate backup should continue if an enqueue failure is encountered.

The Abackup Copy Technique field specifies whether *concurrent copy* should be used during aggregate backup processing.

Valid values are:

P indicates that concurrent copy is preferred and should be used for backup. A data set is backed up on a nonconcurrent copy volume if it does not reside on a volume supported by concurrent copy, or if the volume on which it resides is unavailable for concurrent copy.

R indicates that concurrent copy must be used for backup. Backup fails for data sets that don't reside on volumes supported by concurrent copy, or that are unavailable for concurrent copy.

S indicates standard allocation, in which data sets are backed up without using concurrent copy.

The changes that you make take effect on existing data sets at the next scheduled or requested processing time for those data sets.

Altering Storage Classes

As the needs of your complex change and as new hardware becomes available, you might need to change the storage classes. To modify a storage class, go to the Storage Class Application Selection panel, shown in Figure 30 on page 87, and specify the name of the SCDS containing the storage class you want to change. Specify the name of the storage class and select option 4, Alter.

ISMF displays the Storage Class Alter panel shown in Figure 102.

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

SCDS Name . . . . . : USER8.TEST.SCDS
Storage Class Name  : SC1
To ALTER Storage Class, Specify:
  Description ==> TESTING
              ==>
  Performance Objectives
   Direct Millisecond Response . . . .            (1 to 999 or blank)
   Direct Bias . . . . . . . . . . . .            (R, W or blank)
   Sequential Millisecond Response . .            (1 to 999 or blank)
   Sequential Bias . . . . . . . . . .            (R, W or blank)
   Initial Access Response Seconds . .            (0 to 9999 or blank)
   Sustained Data Rate (MB/sec)  . . .            (0 to 999 or blank)
  Availability . . . . . . . . . . . N            (C, P ,S or N)
  Accessibility  . . . . . . . . . . N            (C, P ,S or N)
   Backup  . . . . . . . . . . . . . .            (Y, N or Blank)
   Versioning  . . . . . . . . . . . .            (Y, N or Blank)

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 102. Altering a Storage Class

This panel contains the same attributes as the Storage Class Define panel. The fields are primed with the information you entered when you defined the storage class. To alter the storage class, type over the old information with your new information on this and the next Storage Class Alter panel. You can use the DOWN command to reach the next panel and use the END command to save the changes and return to the Storage Class Application Selection panel. The changes take effect when you activate this updated configuration.

The changes that you make take effect on existing data sets or objects at the next scheduled or requested processing time for those data sets or objects (for example OSMC cycle for objects).

Altering Data Classes

As the needs of your installation change, you might need to change data classes. To modify a data class, go to the Data Class Application Selection panel, shown in Figure 36 on page 112, and specify the name of the SCDS containing the data class you want to change. Specify the data class name and select option 4, Alter.

ISMF displays the Data Class Alter panels shown in Figure 103, Figure 104 on page 205, Figure 105 on page 205 and Figure 106 on page 206.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              DATA CLASS ALTER                 Page 1 of 4
Command ===>

SCDS Name  . . : SMS.SCDS1.SCDS
Data Class Name: DATAV

To ALTER Data Class, Specify:
  Description ==>
              ==>
  Recfm . . . . . . . . . VB            (any valid RECFM combination or blank)
  Lrecl . . . . . . . . . 255           (1 to 32761 or blank)
  Space Avgrec  . . . . . U             (U, K, M or blank)
        Avg Value . . . . 255           (0 to 65535 or blank)
        Primary . . . . . 5000          (0 to 999999 or blank)
        Secondary . . . . 5000          (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 59 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 103. Data Class Alter Panel, Page 1 of 4

Press Enter to see Page 2 of the Data Class Alter panel, as shown in Figure 104 on page 205.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                              DATA CLASS ALTER            Page 2 of 4
Command ===>

SCDS Name . . . : Y421252.MYSCDS
Data Class Name : DCA11

To ALTER Data Class, Specify:

   Data Set Name Type  . . . . . .      (EXT, HFS, LIB, PDS or blank)
     If Ext  . . . . . . . . . . .      (P=Preferred, R=Required or blank)
     Extended Addressability . . . N    (Y or N)
     Record Access Bias  . . . . .      (S=System, U=User or blank)
   Space Constraint Relief . . . . N    (Y or N)
     Reduce Space Up To (%)  . . .      (0 to 99 or blank)
     Dynamic Volume Count  . . . .      (1 to 59 or blank)
  Compaction . . . . . . . . . . .      (Y, N, T, G or blank)
  Spanned / Nonspanned . . . . . .      (S=Spanned, N=Nonspanned or blank)

  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 104. Data Class Alter Panel, Page 2 of 4

Press Enter to see Page 3 of the Data Class Alter panel, as shown in Figure 105 .

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                              DATA CLASS ALTER             Page 3 of 4
Command ===>

SCDS Name . . . : Y421252.MYSCDS
Data Class Name : DCA11


To ALTER Data Class, Specify:

   Media Interchange
    Media Type  . . . . . . . . . . .     (1, 2, 3, 4 or blank)
    Recording Technology  . . . . . . N   (18, 36, 128, 256 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  . . . . . . . . . .     (1 to 100 or blank)
              CA  . . . . . . . . . .     (1 to 100 or blank)

Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit
```

Figure 105. Data Class Alter Panel, Page 3 of 4

Press Enter to see Page 4 of the Data Class Alter panel, as shown in Figure 106 on page 206.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                              DATA CLASS ALTER             Page 4 of 4
Command ===>

SCDS Name . . . : Y421252.MYSCDS
Data Class Name : DCA11


To ALTER Data Class, Specify:

  Shareoptions Xregion. . . . . . . .      (1 to 4 or blank)
               Xsystem  . . . . . . .      (3, 4 or blank)
  Reuse . . . . . . . . . . . . . . . N    (Y or N)
  Initial Load  . . . . . . . . . . . R    (S=Speed, R=Recovery or blank)
  BWO . . . . . . . . . . . . . . . .      (TC=TYPECICS, TI=TYPEIMS, NO or blank)
  Log . . . . . . . . . . . . . . . .      (N=NONE, U=UNDO, A=ALL or blank)
  Logstream Id  . . . . . . . . . . .      ____________________
  FRlog . . . . . . . . . . . . . . . N    (N=NONE, R=REDO or blank)
  RLS CF Cache Value  . . . . . . . . A    (A=ALL, N=NONE, U=UPDATESONLY)

Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit
```

Figure 106. Data Class Alter Panel, Page 4 of 4

These panels contain the same attributes as the Data Class Define panels. The fields are primed with the information you entered when you defined the data class. To alter the data class, type over the old information with your new information on this and the next panel. You can use the DOWN command to reach the next panel and use the END command to save the changes, and return to the Data Class Application Selection panel. The changes take effect when you activate this updated configuration.

The changes you make apply only to new data sets. Existing data sets are not affected.

In addition, page 1 of this panel has been modified to allow storage administrators to specify a value for the Additional Volume Amount attribute. Additional Volume Amount is applicable only to extended format VSAM multivolume data sets. Additional Volume Amount is specified when Data Set Name Type = EXT and Volume Count>1. The valid values for this attribute are ’P’ (primary), ’S’ (secondary), and blank (not specified; the system default is primary).

Altering Aggregate Groups

As the needs of your installation change, you might need to change aggregate groups. To modify an Aggregate Group, go to the Aggregate Group Application Selection panel, shown in Figure 41 on page 131, and specify the name of the SCDS containing the aggregate group you want to alter. Specify the aggregate group name and select option 4, Alter.

ISMF displays the Aggregate Group Alter panel shown in Figure 107 on page 207.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                            AGGREGATE GROUP ALTER              Page 1 of 2
Command ===>

SCDS Name  . . . . . : SMS.SCDS1.SCDS
Aggregate Group Name : TESTAG

To DEFINE Aggregate Group, Specify:
 Description ==>
             ==>

 Backup Attributes
   Number of Copies  . . . . . . . 1                (1 to 15)
   Management Class Name . . . . .                  (1 to 8 characters, to be
                                                    defined in current SCDS)

 Output Data Set Name Prefix . . .
                                                    (1 to 33 Characters)
 Account . . . . . . . . . . . . .
                                                    (1 to 32 Characters)

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

Figure 107. Aggregate Group Alter, Page 1 of 2

The Aggregate Group Alter panels contain the same attributes as the Aggregate Group Define panels discussed in “Defining Aggregate Group Attributes” on page 131. The fields are primed with the information you entered when defining the aggregate group. To alter the aggregate group, simply type over the old information with your new information on this and the next panel. Use the END command to save the changes.

The first page of the Aggregate Group Alter panel contains aggregate group alter attributes. The SCDS Name and Aggregate Group Name fields are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the aggregate group.

You can specify the following *required* attributes on the first page of the Aggregate Group Alter panel:

Number of Copies
: specifies the number of aggregate backup output files to be altered. The valid values are 1 to 15.

Management Class Name
: specifies the management class name from which the aggregate backup attributes are obtained. The valid values are 1 to 8 alphanumeric characters (first character not a digit) or a blank.

Editing Aggregate Group Attributes

After specifying your backup attribute values, issue the DOWN command to view the second page of the Aggregate Group Alter panel, shown in Figure 108 on page 208.

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
                            AGGREGATE GROUP ALTER               Page 2 of 2
Command ===>


SCDS Name  . . . . . : SMS.SCDS1.SCDS
Aggregate Group Name : AGUSER03
To Edit a Data Set, Specify Number  . .     (1, 2, 3, 4, 5, or 6)
  Selection Data Sets:            (1 to 44 characters)
   1 ===>
     Member Name ===>             (1 to 8 characters)
   2 ===>
     Member Name ===>             (1 to 8 characters)
   3 ===>
     Member Name ===>             (1 to 8 characters)
   4 ===>
     Member Name ===>             (1 to 8 characters)
   5 ===>
     Member Name ===>             (1 to 8 characters)

  Instruction Data Set:           (1 to 44 characters)
   6 ===>
Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 108. Aggregate Group Alter, Page 2 of 2

The second page of the Aggregate Group Define panel contains the selection and instruction data set names for the aggregate group. SCDS Name and Aggregate Group Name are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel.

Edit a Data Set
: Select the number of a selection or instruction data set that you want to edit. When you select a data set number, you can alter the data set by invoking PDF Edit. The PDF edit screen is shown in Figure 44 on page 134. See *z/OS ISPF User's Guide Volume I* for more information on PDF Edit commands.

Selection Data Set Name
: Name of the data set containing lists of data sets to be included in the application backup. You can specify up to five selection data set names. One data set name is required. There is no default. If you want to enter a fully qualified data set name, enclose the name in single quotes. If you do not enclose the name in single quotes, the TSO prefix is added to the name as the first high level qualifier. There is no default.

Member Name
: Name of the data set member containing lists of data sets to be included in the application backup if the selection data set is partitioned. This name must be a valid TSO data set member name. If the data set specified in the Selection Data Set Name field is a partitioned data set, you must specify a valid member name. There is no default.

Instruction Data Set Name
: Name of the data set containing instruction, commands, etc., that are copied into the control file volume after the backup control file. This data set can only be a sequential data set. You must use a valid TSO data set name. The data set name, including the TSO prefix, can be no more than 44 characters long. This is an optional field and has no default.

The changes take effect when you activate this updated configuration.

Altering the SCDS on Different DFSMS Releases

The SCDS can be updated on all supported DFSMS releases. Before updating the SCDS, you must first apply coexistence PTFs on your system to ensure the integrity of the SCDS. Because new functional enhancements are not normally retrofitted, you need to update the SCDS from the most current level of the system. You should always back up the SCDS and ACDS just like any other production data set for the purpose of recovery. See *z/OS DFSMS Migration* for more information.

Copying SMS Classes, Storage Groups, and Aggregate Groups

To simplify the creation of new SMS classes, storage groups, aggregate groups, and SCDSs, you can copy existing ones and modify them. In the LINE OPERATOR column of any List panel, you can enter the COPY line operator and press Enter to duplicate an existing SMS class, storage group or aggregate group. You receive the Copy Entry panel shown in Figure 109.

Tip: Generate your list from an SCDS before issuing a COPY command. Note also that for storage groups, the volume list associated with the storage group is not copied.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                                COPY ENTRY PANEL
Command ===>

Construct will be Copied from:

  Data Set Name . . : SMS.SCDS1.SCDS
  Construct Name  . : DATAV
  Construct Type  . : MANAGEMENT CLASS

Specify "Copy To" Construct:

  Data Set Name . . . 'SMS.SCDS2.SCDS'
                               (1 to 46 Characters)

  Construct Name  . .          (1 to 8 characters, fully specified)

Enter "/" to select option   _  Replace like-named Construct

                             _  Perform Alter

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

Figure 109. Copy Entry Panel

The *from* Data Set Name field identifies the source you are copying. It is primed with the value you specified on the Application Selection panel. The *from* Construct Name field identifies the name of the SMS class or storage group you are copying. It is primed with the value from the Name field of the List panel. The *from* Construct Type field identifies whether you are copying a data class, storage class, management class, aggregate group or storage group. It is primed with the type of List panel from which you issued the COPY command. You should generate your list from an SCDS instead of the ACDS prior to copying SMS classes, storage groups, and aggregate groups.

The *to* Data Set Name field identifies the target of the copy. It must be the name of an SCDS. It is primed with the value of the *from* Data Set Name if the *from* Data Set Name contains an SCDS name. The *to* Construct Name field identifies the name of the SMS class or storage group copy. It is primed with blanks. The copy has the same *type* as the original.

If the SMS class, storage group, or aggregate group that you identify in the *to* fields already exists, you specify whether you want to replace it in the Replace like-named Construct field. If you indicate Y, yes, the replacement occurs. If you indicate N, no, the copy does not occur. The field is primed with N, which is the default.

In the Perform Alter field, you indicate if you want to change some of the attributes of the copy you are creating. If you specify Y, yes, the system displays the Management Class Application Selection panel. This panel lists four groups of management class attributes. You can then choose an attribute group for altering ahead of other attributes. If you specify N, no, you remain on the Copy Entry panel, where you can perform another copy or return to the original List panel. The field is primed with N, which is the default.

When you have specified the values, press Enter to perform the copy.

Deleting DASD Volumes from the System

You might want to delete DASD volumes from the system. You can do this without performing an initial program load (IPL) by following this procedure:

1. Set the DASD volume status to DISNEW. This disables any new allocations to the volume.
2. Move data sets that you want to keep to other volumes using DFSMSdss.
3. Delete the DASD volume from the storage group that contains it. See “Deleting DASD Volumes from Storage Groups” for information on deleting DASD volumes.
4. Vary the DASD volume offline. See *z/OS DFSMShsm Storage Administration Guide* if you need specific information about how to vary a volume offline.
5. Issue the DFSMShsm DELVOL command from all DFSMShsm systems that are aware of the volume. This prevents DFSMShsm from attempting to allocate to the volume during subsequent interval migrations. See *z/OS DFSMShsm Storage Administration Guide* if you need specific information on the DFSMShsm DELVOL command.

Deleting DASD Volumes from Storage Groups

When you delete a DASD volume from a storage group, first set the volume status to DISNEW. This prevents any new allocations to the volume while allowing existing data sets to still be accessed with DISP=SHR, DISP=MOD, or DISP=OLD. Move any data that you do not want converted to non-SMS from the volume. Next, if the DASD volume is to be reused as a non-SMS-managed volume, run a DFSMSdss NONSMS CONVERTV to convert it out of SMS. Finally, remove the volume from the storage group.

To remove DASD volumes from a storage group, select the Storage Group Application Selection panel. Specify the CDS name and the storage group name that contains the volumes you want to delete. Press Enter to get the Storage Group Volume Selection panel shown in Figure 100 on page 201. From the Storage Group Volume Selection panel, indicate the volumes you want to delete in the SPECIFY A SINGLE VOLUME (in PREFIX), OR RANGE OF VOLUMES field, select option 4,

and press Enter. Be careful when moving or removing a volume from a storage group because the volume could contain part of a multivolume data set. The changes take effect when you activate this updated configuration.

To prevent DFSMShsm from attempting to allocate to volumes which have been deleted from the storage group, issue the DFSMShsm DELVOL command from all DFSMShsm systems which are aware of the deleted volume.

After deletion, the DASD volume is only eligible for non-SMS allocations. However, if you are reassigning the volume to another storage group, you need to define the volume to the storage group and then activate the updated configuration.

Deleting Storage Groups

You might want to delete a pool type storage group so that you can define all of its volumes to other storage groups or so that you can use the volumes for non-SMS allocations. The following sections describe the procedure for deleting storage groups and some considerations for preventing job failures.

Deleting or Moving System-Managed Data Sets from the Storage Group

If you plan to use the volumes for non-SMS allocations, you need to delete or move all system-managed data sets. The following procedure shows how to do this and how to remove the storage group from the SMS configuration:

1. Set the storage group status to DISNEW to prevent any further new allocations to volumes defined to the storage group. As time passes, DFSMShsm space management processing clears most of the data sets off the volumes in the storage group you are deleting, provided AUTO MIGRATE is ‘Y’ and most of the data sets have an associated management class.
2. Use DFSMSdss to move any remaining data sets that have lingered on the storage group volumes when the volumes are sufficiently empty.
3. Build an SCDS to contain the new configuration. Remove the storage group definition from the SCDS and change the storage group ACS routine so that it does not select the deleted storage group.
4. Activate the new configuration.

To declare volumes non-SMS-managed, you need to INIT them through ICKDSF to non-SMS or CONVERT them through DFSMSdss to non-SMS.

When you delete a VIO or dummy type storage group, you do not have to worry about redefining volumes to other storage groups, but you might need to modify the storage group ACS routine.

Preventing Job Failures

You should be aware that if you delete a storage group while a job that uses this group is running, the job might fail with message IGD17201I. To prevent this, consider performing the following:

1. Modify your storage group ACS routine so that it does not assign the storage group to be deleted to any new allocations. Validating your configuration causes message IGD06023I to be received. This message is only a warning. As long as there are no error messages, your configuration is valid.

2. Activate the configuration with the modified storage group ACS routine and allow it to remain active until jobs that used the old ACS routine have completed. This allows executing jobs to complete using the existing storage group definition.
3. Delete the storage group from the configuration. Since the storage group no longer exists in the configuration, you do not receive MSGIGD06023I.
4. Activate the configuration from which the storage group has been deleted.

Deleting SMS Classes

You need to exercise caution when you delete storage, management and data classes, because a user or system might attempt to use a data set that references them.

For example, assume a data set has some specified migration attributes, and at some point it meets the migration criteria. After the data set has been migrated, you delete its associated management class. If the data set is recalled in the future, it must go through the management class ACS routine. If the ACS routine fails to override the undefined or deleted management class, the recall fails because a management class that does not exist is associated with the data set. If this occurs, you must either rewrite the management class ACS routine or specify that the data set should be recalled as a non-SMS data set. If you rewrite the ACS routine, you must check for the deleted management class and assign a valid management class to the data set. If you decide to bypass ACS processing, be aware that the data set is recalled to non-SMS storage.

Another consequence of deleting management classes involves DFSMShsm automatic processing. When DFSMShsm runs through its automatic cycle and processes data sets on the basis of their management class attributes, it attempts to retrieve nonexistent management class definitions. Consequently, DFSMShsm skips the processing of these data sets.

Instead of deleting storage classes and management classes, you should prevent any new allocation from using them by rewriting the corresponding ACS routine to override the deleted storage class or management class. It is safer and provides the possibility that all data sets that reference the management class or storage class might eventually be overridden. Also, you can identify all data sets with storage class or management class and then use ALTER to change to a new storage class or management class. If you decide to delete a storage class or management class, make certain that you inform all users well in advance.

Because data classes are used only at data set allocation, they do not have these problems. The original data class name is never referenced or reused.

If you decide to delete an SMS class, enter the following information on the pertinent Class Application Selection panel:

```
SCDS NAME         ===> 'SMS.SCDS1.SCDS'
xxxx CLASS NAME   ===> *
SELECT OPTION     ===> 1
```

After you press Enter, ISMF displays the Class List panel. Next to the SMS class that you want to delete, enter DELETE in the LINE OPERATOR column. When you press Enter, you see the Confirm Delete Request panel shown in Figure 110 on page 213.

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                              CONFIRM DELETE REQUEST
Command ===>

  To Confirm Deletion on the following element:

     Element Name . . : SYS4
     Element Type . . : SYS GROUP NAME
     Residing in SCDS : SMS.SCDS1.SCDS


  Specify the following:
      Enter "/" to select option   /  Perform Deletion









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

Figure 110. Confirm Delete Request Panel

Confirm that the displayed SMS class is the one that you want to delete. If it is, enter Y for yes and press Enter. The SMS Class List should appear with '*DELETE' in the LINE OPERATOR column next to the deleted SMS class.

After deleting SMS classes, activate the modified SMS configuration to make your changes part of the active configuration.

Converting Volumes to SMS

You can convert empty volumes through ISMF using the INIT line operator from the Volume application. You can convert previously-used, non-SMS-managed volumes using the CONVERTV command of DFSMSdss. You can also unconvert SMS volumes using the CONVERTV command of DFSMSdss.

See *z/OS DFSMS: Implementing System-Managed Storage* for information on how to convert and unconvert volumes.

Chapter 12. Recovering Storage Management Subsystem Information

This chapter describes how you can recover SMS information, including control data sets, source control data sets (SCDS), active control data sets (ACDS), communications data sets (COMMDS), and the SMS address space.

Recovering Control Data Sets

Allocate a spare ACDS and COMMDS when you allocate your originals. Having spare copies eases the recovery process if SMS cannot access the originals because of I/O errors. Place the spares on a device that every system in your complex can access. Make certain that they reside on a different device from your originals.

In order to support 32 names, an SMS control data set is converted from 8-name mode to 32-name mode. You must confirm this conversion, using the operator console or via ISMF.

Restrictions and Recommendations:

- You cannot use an ACDS as an SCDS if the SCDS is lost, because an ACDS cannot be defined or altered.
- If an SMS system temporarily goes down or has not yet been started with SMS, then it is not part of the system-managed storage environment. When you activate SMS on the system, it uses the ACDS and COMMDS that are specified in its IGDSMSxx member. If the COMMDS and ACDS are different from the ones used by the remainder of the SMS complex, then the system runs as a separate SMS complex.

  Consequently, when you change the current ACDS or COMMDS, update the IGDSMSxx member of SYS1.PARMLIB for each system in the SMS complex.
- Make copies of your control data sets before the system mode is converted for the following reasons:
  – The conversion is permanent.
  – You might want to maintain them for coexistence with down-level systems.
- There are special considerations to keep in mind when you move control data sets and their catalog entries. See *z/OS DFSMS: Managing Catalogs* for more information on special considerations, including the use of REPRO MERGECAT.

Recovering an SCDS

For purposes of recovery, treat an SCDS the same as any other source VSAM linear data set. You can use DFSMShsm to manage its availability, which relieves you from having to allocate any spares.

Recovering an ACDS

You can recover from errors that prevent access to the ACDS if you have allocated a spare. All permanent errors that make the ACDS unreadable or unwritable require intervention. For permanent I/O errors to the ACDS, the messages IGD041I and IGD040D appear on the operator console. You have the option of retrying the failing operation or using the spare ACDS.

If you decide to use the spare ACDS, you need to perform three steps.

1. Reply 'C' on the operator console.

© Copyright IBM Corp. 1986, 2002

2. Copy the data from the SMS address space to the spare ACDS by issuing the following command from a system in the SMS complex:

   ```
   SETSMS SAVEACDS(spare.acds)
   ```

3. Tell the system to use the spare ACDS by issuing the following command:

   ```
   SETSMS ACDS(spare.acds)
   ```

You need only to issue the `SETSMS ACDS(spare.acds)` command on one system. The COMMDS is updated to reflect this change. As the other systems in the SMS complex access the COMMDS (based on the INTERVAL value in their respective IGDSMSxx members), they automatically switch to the new ACDS.

Recovering a COMMDS

If you cannot access the COMMDS, you can recover from the error if you allocated a spare data set. All permanent errors that make the COMMDS unreadable require intervention. For permanent I/O errors to the COMMDS, the messages IGD041I and IGD070D appear on an operator console. Reply 'S' on the operator console and issue the following command from a system in the SMS complex:

```
SETSMS COMMDS(spare.commds)
```

One of three situations results.

1. If the spare COMMDS is empty it gets formatted automatically, and SMS writes the in-storage copy of the current COMMDS into the `spare.commds`. You then need to issue the following command on each of the remaining systems in the SMS complex:

   ```
   SETSMS COMMDS(spare.commds)
   ```

2. If the spare COMMDS is not empty but describes an ACDS that is not currently active in the SMS complex, then SMS issues the message IGD076D. This message asks if you want to use the contents of the COMMDS and the ACDS to which it points. Reply 'C' to cause SMS to replace the contents of the `spare.commds` with the in-storage copy of the current COMMDS. You then need to issue the following command on each of the remaining systems in the SMS complex:

   ```
   SETSMS COMMDS(spare.commds)
   ```

3. If the spare COMMDS is not empty but describes the ACDS that is currently active in the SMS complex, you need to issue the following command on each of the remaining systems in the SMS complex:

   ```
   SETSMS COMMDS(spare.commds)
   ```

A response of ’S’ to IGD070D is recommended when recovering from the current COMMDS because a response of ’C’ might result in an unrecoverable error when trying to reaccess the current COMMDS. When access to the current COMMDS is suspended, SMS is able to access the new COMMDS without accessing the current COMMDS and resulting in further errors.

Without a usable COMMDS, the systems in the SMS complex have no means of communication. Other systems in the SMS complex are aware of the error, but they are unaware of the switch to a new COMMDS until you inform them.

If you can access the current COMMDS but you want to use an alternate one, you only need to issue the `SETSMS` command from one system. The other systems in the SMS complex detect the change from the old COMMDS to the new COMMDS and they automatically switch to the new one.

Recovering from a Systems Failure in the SMS Complex

To provide for recovery procedures, you should leave one system or system group name slots empty in the corresponding function panels. In case a problem occurs requiring that a system that is defined within the configuration as part of a system group be specified by its specific name (i.e., hardware failure), you can add it without disrupting the rest of the system group of SMS complex.

You can initiate recovery with the following steps:

- Use the ISMF CDS Application to add the system name to the SMS configuration.
- Activate the configuration, either from ISMF or from the operator console.
- Use the VARY SMS operator command to make any necessary changes to the configuration.
- Remove the system name from the configuration and reactivate the configuration once the problem has been corrected.

Recovering the SMS Address Space

If the SMS address space fails, SMS automatically attempts to restart up to six times. If SMS fails to recover its address space after the sixth restart, you have two options:

1. You can end the SMS address space and then restart SMS using the `T SMS=xx` command. If you have cross memory interactions, users' address spaces wait for the SMS address space to restart. After the sixth automatic restart, SMS displays a message requesting the user to select the next action; retry another restart or terminate the SMS address space. This relates to the SMS address space and *not* the user's address space.
2. You can allow SMS to attempt another restart by replying "YES" to the system-generated message:

   `IGD032D`

Canceling the SMS Address Space

You cannot cancel SMS once it is activated. You must re-IPL to cancel the SMS address space and deactivate SMS. If the system is set up with automatic activation of SMS at IPL time and you want to deactivate SMS, you need to modify the appropriate SYS1.PARMLIB members to prevent automatic activation of SMS.

Chapter 13. Protecting the Storage Management Subsystem

To ensure that SMS operates correctly, you need to prevent unauthorized end users from modifying information in certain control data sets, such as SMS class, aggregate group, and storage group definitions. Some ISMF commands and functions also require protection from use by unauthorized users. This chapter explains how to use Resource Access Control Facility (RACF), a component of the Security Server for z/OS, to establish authorization levels for protecting these data sets, commands, and functions.

Data set password protection does not apply to SMS-managed data sets or catalogs by SMS. SMS assumes the presence of RACF or an equivalent security product. Password protection is bypassed for all system-managed data sets.

Identifying the Resource Owner and Extracting the Default Classes

The *resource owner* is the actual owner of the data set covered by the RACF DATASET profile. RACF extracts the resource owner based on data set name. If the resource owner is not identified, the high-level qualifier is used.

If you specify ACSDEFAULTS=YES in the IGDSMSxx member, RACF uses the resource owner to extract the default SMS classes and application identifier. If the resource owner is a user and no default SMS information is available from the user profile, the default information from the group profile is used. If the resource owner is a group, then the defaults for the group profile is used. You can protect the ability to update the resource owner field, RESOWNER, in the data set RACF profile. Revoked USERID should not be used as a resource owner, or it causes RACF to fail. See also “FIELD Resource Class” on page 233.

If you specify USE_RESOWNER=NO in the IGDSMSxx member, RACF uses the execution user ID instead of the resource owner to check authorization. This allows users who do not use a naming convention, user ID or group ID as the high level qualifier of data set names to check authorization to use storage and management classes. If you specify USE_RESOWNER=YES, there is no change to current processing. This is the default.

After ACS routines have been run, RACF is invoked to verify the user’s authority to allocate the data set (CREATE/ALTER) and the resource owner’s authority to use the STORCLAS and MGMTCLAS (READ). You can protect the ability of a resource owner to use management class and storage class through STORCLAS and MGMTCLAS resource classes.

See *MVS/ESA SML: Managing Data* for additional information on using RACF in an SMS environment.

© Copyright IBM Corp. 1986, 2002

Protecting ISMF Functions

You can use RACF authorization to limit access to the following categories of ISMF functions:

1. The entire ISMF component
2. The individual ISMF applications:

Profile	Data Set
DASD Volume	Mountable Optical Volume
Management Class	Data Class
Storage Class	Storage Group
Automatic Class Selection	Control Data Set
Aggregate Group	Library Configuration
Library Configuration	Drive Configuration
Data Collection	List

3. The ISMF line operators
4. The ISMF commands

ISMF relies on the *RACF program control feature* to protect many of its applications. The RACF program control feature prevents unauthorized end users from running selected ISMF programs. To use the feature, you must activate the RACF Program Class and define your selected ISMF programs to RACF.

With RACF program control you can set up authorization levels for each of these categories, varying the level within a particular category to suit the needs of your installation. Individual end users can execute an ISMF function if one of the following conditions is true:

- They are authorized to execute the corresponding load module.
- Their RACF profile contains the OPERATIONS attribute.
- Their group is authorized to execute the load module.
- RACF is disabled or the program control feature is turned off.
- The universal access authority (UACC) for the load module is READ or greater, making the load module available to anyone who can access ISMF.

Recommendation: Protect these functions with RACF program control to make sure that only particular users can use the storage administrator applications and functions. Because a TSO user can change his user mode level, as this information is contained in the user's ISPF profile, protect the functions at a different level than user mode level.

The RACF program resource class allows the security administrator to protect various ISMF applications and functions with program control. This is achieved by controlling the access to load modules which are invoked by:

- ISMF Applications
- ISMF Line Operators
- ISMF Commands

The load modules reside in the following libraries:

- SYS1.DGTLLIB for DFSMSdfp/ISMF
- SYS1.DGTLLIB for DFSMSdss/ISMF
- SYS1.DFQLLIB for DFSMShsm

If the installation moves these modules to another load library, the installation-defined load library must be used in the program protection.

To protect a load module, use the RDEFINE RACF command. The syntax of this command is:

```
RDEFINE PROGRAM mod-name OWNER(owner of profile)          +
        UACC(NONE) ADDMEM('dsn of loadlib'/volser/NOPADCHK)
```

See *z/OS Security Server RACF Security Administrator's Guide* for a detailed description of how to use the RACF program control features.

Locating Module Names for ISMF Applications

Table 14 lists the load module names you can use to limit access to certain ISMF applications. The names are included in the panel coding for the ISMF Primary Option Menu for Storage Administrators, which is stored in the panel library with a member name of DGTSMMD2. If you want to limit access to all of ISMF, use module name DGTFMD01.

Table 14. Module Names for ISMF Applications

Application	Module Name
Profile	DGTFPF00
Data set	DGTFDS00
DASD Volume	DGTFVA00
Mountable Optical Volume	DGTFOVCD
Management class	DGTFMCCD
Data class	DGTFDCCD
Storage class	DGTFSCCD
Storage group	DGTFSGDR
Automatic class selection	DGTFFLAD
Control data set	DGTFSACD
Aggregate Group	DGTFAGCD
Optical Library Configuration	DGTFLCCD
Optical Drive Configuration	DGTFRCCD
Data Collection	DGTFADAD
List	DGTFJLCD
Mountable Tape Volume	DGTFTVCD
Tape Library Configuration	DGTFLMCD

Locating Module Names for ISMF Functions

Restricting access to the DGTFPF05 module prevents end users from gaining access to the Primary Option Menu for Storage Administrators. Table 15 lists this and the other load module names you can use to limit access to certain ISMF functions.

Table 15. Module Names for ISMF Functions

Function	Module Name
User mode	DGTFPF05
Logging and abend control	DGTFPF02
ISMF job statement information	DGTFPF03
DFSMSdss execute statement information	DGTFPF04
ICKDSF execute statement information	DGTFPF20
Data set print execute statement information	DGTFPF21

Table 15. Module Names for ISMF Functions (continued)

Function	Module Name
IDCAMS execute statement information	DGTFPF22
Data Class DEFINE	DGTFDCDA
Data Class ALTER	DGTFDCAA
Data Class DISPLAY	DGTFDCDI
Data Class LIST	DGTFDCLD
Storage Class DEFINE	DGTFSCDA
Storage Class ALTER	DGTFSCAA
Storage Class DISPLAY	DGTFSCDI
Storage Class LIST	DGTFSCLD
Management Class DEFINE	DGTFMCDA
Management Class ALTER	DGTFMCAA
Management Class DISPLAY	DGTFMCDI
Management Class LIST	DGTFMCLD
Storage Group DEFINE	DGTFSGFR
Storage Group ALTER	DGTFSGAR
Storage Group LIST	DGTLGP24
Storage Group VOLUME	DGTFSGVR
Aggregate Group DEFINE	DGTFAGDA
Aggregate Group ALTER	DGTFAGAA
Aggregate Group DIQPLAY	DGTFAGDI
Aggregate Group LIST	DGTFAGLD
Optical Library Configuration DEFINE	DGTFLCDE
Optical Library Configuration ALTER	DGTFLCAL
Optical Library Configuration DISPLAY	DGTFLCDI
Optical Library Configuration LIST	DGTFLCLD
Optical Drive Configuration DEFINE	DGTFRCDE
Optical Drive Configuration ALTER	DGTFRCAL
Optical Drive Configuration DISPLAY	DGTFRCDI
Optical Drive Configuration LIST	DGTFRCLD
Tape Library Configuration DEFINE	DGTFLMDE
Tape Library Configuration ALTER	DGTFLMAL
Tape Library Configuration DISPLAY	DGTFLMDI
Tape Library Configuration LIST	DGTFLMLD

Locating Module Names for Line Operators and Commands

Table 16 on page 223, Table 17 on page 224, Table 18 on page 225 and Table 19 on page 226 list the module names for ISMF line operators and commands. The names are stored in command tables in the DGTLLIB load library. The line operators for storage administrators are described in “Appendix A. Quick Reference to ISMF Commands and Line Operators” on page 295. Line operators for end users are described in *z/OS DFSMS: Using the Interactive Storage Management Facility*.

Tip: You can invoke CATLIST or VTOCLIST from the Data Set List or outside of ISMF, but they are not ISMF line operators with ISMF load module names.

Table 16. Module/CLIST Names for ISMF Line Operators, Part I

Line Operator	Aggregate Group List	Data Class	Data Set List	Optical Drive List
ALTER	DGTFALH1	DGTFALD1	DGTFAL01	DGTFALR1
ANALYZE	-	-	-	-
BROWSE	-	-	DGTFBR01	-
CLIST	-	-	DGTFCL01	-
COMPRESS	-	-	DGTFCM01	-
CONDENSE	-	-	DFQFCND1	-
CONVERTV	-	-	-	-
COPY	DGTFCAH1	DGTFCAD1	DGTFCY01	DGTFCAR1
DEFRAG	-	-	-	-
DELETE	DGTFDNH1	DGTFDND1	DGTFDL01	DGTFDNR1
DISPLAY	DGTFDIH1	DGTFDID1	-	DGTFDIR1
DUMP	-	-	DGTFDP01	-
EDIT	-	-	DGTFED01	-
EJECT	-	-	-	-
ERASE	DGTFDNH1	DGTFDND1	DGTFDL01	DGTFDNR1
HALTERDS	-	-	DFQFHA01	-
HBACKDS	-	-	DFQFHB01	-
HBDELETE	-	-	DFQFHBD1	-
HDELETE	-	-	DFQFHD01	-
HIDE	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01
HMIGRATE	-	-	DFQFHM01	-
HRECALL	-	-	DFQFHRL1	-
HRECOVER	-	-	DFQFHRC1	-
INIT	-	-	-	-
INSPECT	-	-	-	-
LIST	-	-	-	-
LISTSYS	-	-	-	-
LISTVOL	-	-	-	-
MESSAGE	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00
RAUTH	-	-	-	-
RECOVER	-	-	-	-
RELEASE	-	-	DGTFRL01	-
RESTORE	-	-	DGTFRT01	-
REFORMAT	-	-	-	-
SECURITY	DGTFSRD1	-	DGTFSRD1	-
SETCACHE	-	-	-	-
STATUS	-	-	-	-
TSO Commands and CLIST	DGTFUS01	DGTFUS01	DGTFUS01	DGTFUS01

Table 17. Module/CLIST Names for ISMF Line Operators, Part 2

Line Operator	Optical Library List	List	Management Class	Mountable Optical Volume List
ALTER	DGTFALL1	-	DGTFALM1	-
AUDIT	DGTFAUL1	-	-	DGTFAU01
ANALYZE	-	-	-	-
BROWSE	-	-	-	-
CLIST	-	-	-	DGTFCL01
COMPRESS	-	-	-	-
CONDENSE	-	-	-	-
CONVERTV	-	-	-	-
COPY	DGTFCAL1	-	DGTFCAM1	-
DEFRAG	-	-	-	-
DELETE	DGTFDNL1	DGTFEL01	DGTFDNM1	-
DISPLAY	DGTFDIL1	-	DGTFDIM1	-
DUMP	-	-	-	-
EDIT	-	-	-	-
EJECT	-	-	-	DGTFEF01
ERASE	DGTFDNL1	DGTFEL01	DGTFDNM1	-
HALTERDS	-	-	-	-
HBACKDS	-	-	-	-
HBDELETE	-	-	-	-
HDELETE	-	-	-	-
HIDE	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01
HMIGRATE	-	-	-	-
HRECALL	-	-	-	-
HRECOVER	-	-	-	-
INIT	-	-	-	-
INSPECT	-	-	-	-
LIST	-	DGTFLL01	-	-
LISTSYS	-	-	-	-
LISTVOL	DGTFLVL1	-	-	-
MESSAGE	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00
RAUTH	-	-	-	-
RECOVER	-	-	-	DGTFRC01
RELEASE	-	-	-	-
REMAP	DGTFRML1	-	-	-
RESTORE	-	-	-	-
REFORMAT	-	-	-	-
SECURITY	-	-	DGTFSRD1	-
SETCACHE	-	-	-	-
STATUS	-	-	-	-
TSO Commands and CLIST	DGTFUS01	DGTFUS01	DGTFUS01	DGTFUS01

Table 18. Module/CLIST Names for ISMF Line Operators, Part 3

Line Operator	Storage Class	Storage Group	Volume	Tape Library	Mountable Tape Volume
ALTER	DGTFALS1	DGTFALG1	-	DGTFALY1	DGTFAL11
ANALYZE	-	DGTFAZ01	DGTFAZ01	-	-
AUDIT	-	-	-	DGTFAUL1	DGTFAU04
BROWSE	-	-	-	-	-
BUILDX	-	-	DGTFBX01	-	-
CLIST	-	-	DGTFCL01	DGTFCL01	DGTFCL01
COMPRESS	-	-	DGTFCS01	-	-
CONDENSE	-	-	-	-	-
CONTROL	-	-	DGTFCT01	-	-
CONVERTV	-	-	DGTFCN01	-	-
COPY	DGTFCAS1	DGTFCAG1	DGTFCV01	DGTFCAY1	-
DEFRAG	-	-	DGTFDF01	-	-
DELETE	DGTFDNS1	DGTFDNG1	-	DGTFDNY1	-
DISPLAY	DGTFDIS1	-	-	DGTFDIY1	-
DUMP	-	-	DGTFDM01	-	-
EDIT	-	-	-	-	-
EJECT	-	-	-	-	DGTFEJ01
ERASE	DGTFDNS1	DGTFDNG1	-	-	-
HALTERDS	-	-	-	-	-
HBACKDS	-	-	-	-	-
HBDELETE	-	-	-	-	-
HDELETE	-	-	-	-	-
HIDE	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01
HMIGRATE	-	-	-	-	-
HRECALL	-	-	-	-	-
HRECOVER	-	-	-	-	-
INIT	-	-	DGTFIN01	-	-
INSPECT	-	-	DGTFIV01	-	-
INSTALL	-	-	DGTFIL01	-	-
LIST	-	-	DGTFIV01	-	-
LISTSYS	-	DGTFLIC1	-	-	-
LISTVOL	-	DGTFLVC1	-	DGTFLVL1	-
MESSAGE	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00
RAUTH	-	-	DGTFRA01	-	-
RECOVER	-	-	-	-	-
REFORMAT	-	-	DGTFRF01	-	-
RELEASE	-	-	DGTFRV01	-	-
RESTORE	-	-	DGTFRO01	-	-
REVAL	-	-	DGTFRB01	-	-
SECURITY	DGTFSRD1	-	-	DGTFSRD1	-
SETCACHE	-	-	DGTFCB01	-	-
STATUS	-	-	-	-	
TSO Commands and CLIST	DGTFUS01	DGTFUS01	DGTFUS01	-	-

Table 19. Module/CLIST Names for ISMF Commands

Command	Module Name	CLIST
ACTIVATE	DGTFACAT	-
AUDIT	DGTFAU02	-
BOTTOM	DGTFDO01	-
CLEAR	DGTFCR01	-
COMPRESS	DGTFCP01	-
COPY	DGTFCO01	-
DOWN	DGTFDO01	-
DSUTIL	-	DSUTIL
DUMP	DGTFDU01	-
ERTB	DGTFER02	-
FILTER	DGTFFI01	-
FILTER CLEAR	DGTFFI01	-
FIND	DGTFFN01	-
FOLD	DGTFFU01	-
LEFT	DGTFLE01	-
LIBRARY	-	LIBRARY
LISTPRT	DGTFPR01	-
MIGRATE	-	MIGRATE
PROFILE	DGTFPF01	-
QRETRIEV (DS)	ACBUTO3	-
QRETRIEV (DVOL)	ACBUTO4	-
QSAVE (DS)	ACBUTO6	-
QSAVE (DVOL)	ACBUTO7	-
RECALL	-	RECALL
RECOVER	DGTFRC01	-
RELEASE	DGTFRE01	-
RESHOW	DGTFRW01	-
RESTORE	DGTFRR00	-
RIGHT	DGTFRI01	-
SAVE	DGTFSLDS	-
SORT	DGTFSO01	-
TOP	DGTFUP01	-
UP	DGTFUP01	-
VALIDATE	DGTFVLVA	-
VIEW	DGTFVW01	-

If you want to look at the command tables, you need the data set name that your installation uses for the load library. The install process for ISMF puts the load modules in SYS1.DGTLLIB (for DFSMSdfp/ISMF and DFSMSdss/ISMF), and SYS1.DFQLLIB (for DFSMShsm/ISMF). However, your installation's post-install procedures might involve moving the ISMF modules to other libraries. If the libraries have moved but have kept the same names, you can determine library data sets names by issuing the TSO/E LISTALC command and scanning the low-level qualifiers for DGTLLIB and DFQLLIB.

Protecting Modules

Take the following three program control steps to protect modules:

1. Use the RDEFINE command or the RACF ISPF entry panels to identify the modules you want to protect. To define the modules to RACF, supply the name of the load module that you want to protect, the name of the data set that contains the load module, and the volume serial number of the volume that contains the data set. RACF adds each module that you identify to the profile for the PROGRAM general resource class.

   When you define the modules, you have several options:

   - If you want to define several modules at the same time, you can use asterisk notation. For example, DGT* represents all of the modules beginning with the letters DGT.
   - You can add an access list with user IDs or group names and their associated access authority to the profile.
   - You can define the UACC to give default access to all users or to none.
   - You can use the AUDIT parameter to set up or to bypass RACF logging.
2. Use the PERMIT command to allow end users to execute an application, line operator, or command associated with a module.
3. To prevent unauthorized users from copying a program, renaming it to a name that is unknown to the program control, and then executing the renamed program, you should protect the PDS libraries containing RACF-controlled programs with a UACC of NONE. In order for users to execute programs in these libraries, place the libraries in the LNKLIST concatenation. See *z/OS Security Server RACF Security Administrator's Guide* for more information.

Storage Administration (STGADMIN) Profiles in the FACILITY Class

In order to control the ability to perform functions associated with storage management, define profiles in the FACILITY class whose profile names begin with STGADMIN (storage administration). Although some of these FACILITY profiles are used to protect the Storage Management Subsystem, they are checked whether or not you are using SMS.

There are two sets of facility class profiles. The first are command and keyword related and the second are oriented towards actions and provide authority for storage administrators to do things like backup or copy data sets to which the administrator would usually have no authority.

Command and Keyword Related Profiles

The RACF FACILITY resource class can control the ability to:

- Activate a configuration
- Perform certain catalog functions on data sets using access method services
- Perform certain DFSMSdss functions on data sets

If defined, these profiles are checked before a user is allowed to perform the protected function. If these profiles are not defined, other RACF or password checking is still made to verify authority. Also, the user program must be Authorized Program Facility (APF)-authorized.

Exception: Password checking is bypassed if the function is performed on a system-managed data set.

Some FACILITY profiles are not checked if the caller is using the system key or is running in supervisor state. These profiles are:

STGADMIN.IGG.DEFNVSAM.NOBCS
STGADMIN.IGG.DEFNVSAM.NONVR
STGADMIN.IGG.DELETE.NOSCRTCH
STGADMIN.IGG.DELGDG.FORCE
STGADMIN.IGG.DELNVR.NOBCSCHK
STGADMIN.IGG.DIRCAT

In addition to the individual profiles, we recommend that the STGADMIN.* profile be defined with UACC NONE. Some STGADMIN profiles allow you to perform a specific function or use a specific keyword, but some functions or keywords can be used unless there is a STGADMIN profile preventing the usage of that keyword. By defining an STGADMIN.* profile with UACC NONE, these sensitive keywords can be protected and system exposures eliminated.

You need to define the following RACF profiles:

STGADMIN.DPDSRN.olddsname
Controls the ability to rename a non-SMS-managed data set whose name is in use in another address space. You can regard them as having duplicate data set names.

olddsname is up to 23 characters of the existing data set name. You can use a generic class name such as STGADMIN.DPDSRN.SYS2.*.

Recommendation: Do not give anyone authority to STGADMIN.DPDSRN.* because it is too broad. This option should be used with extreme caution. Very few people should have RACF authority to STGADMIN.DPDSRN.olddsname. Use this option only if you know the data set is not open on any system. For details of how to use this, see *z/OS DFSMSdfp Advanced Services*.

STGADMIN.IDC.BINDDATA
Controls the ability to use the access method services BINDDATA command.

STGADMIN.IDC.DCOLLECT
Controls the ability to use the access method services DCOLLECT command.

STGADMIN.IDC.DIAGNOSE.CATALOG
Controls the ability to run the access method services DIAGNOSE command against catalogs.

STGADMIN.IDC.DIAGNOSE.VVDS
Controls the ability to run the access method services DIAGNOSE command against a VVDS when a comparison against the BCS is performed. In this case, the BCS is protected.

STGADMIN.IDC.EXAMINE.DATASET
Controls the ability to run the access method services EXAMINE command against integrated catalog facility catalog data sets.

STGADMIN.IDC.LISTDATA
Controls the ability to use the access method services LISTDATA command.

STGADMIN.IDC.LISTDATA.ACCESSCODE
Controls the ability to use the access method services LISTDATA ACCESSCODE command. ACCESSCODE is a specialized LISTDATA command that requires an extra level of protection. Both levels (LISTDATA and LISTDATA.ACCESSCODE) are required.

STGADMIN.IDC.SETCACHE
Controls the ability to use the access method services SETCACHE command. This RACF profile does not include the following four profiles. A user must have SETCACHE access in order to have specific setcache command authorization.

STGADMIN.IDC.SETCACHE.DISCARDPINNED
Controls the ability to use the access method services SETCACHE DISCARDPINNED command.

STGADMIN.IDC.SETCACHE.PENDINGOFF
Controls the ability to use the access method services SETCACHE PENDINGOFF command.

STGADMIN.IDC.SETCACHE.REINITIALIZE
Controls the ability to use the access method services SETCACHE REINITIALIZE command.

STGADMIN.IDC.SETCACHE.SUBSYSTEM
Controls the ability to use the access method services SETCACHE SUBSYSTEM command.

STGADMIN.IGD.ACTIVATE.CONFIGURATION
Controls the ability to activate an SMS configuration.

STGADMIN.IGG.ALTBCS
Controls the ability to alter BCS catalog attributes.

STGADMIN.IGG.ALTER.SMS
Controls the ability to alter the storage class and management class of a data set. If this profile is not created, the user must have RACF authority to the storage class and the management class in order to alter it.

STGADMIN.IGG.ALTER.UNCONVRT
Controls the ability to alter a system-managed VSAM data set to an unmanaged VSAM data set.

STGADMIN.IGG.DEFDEL.UALIAS
Controls the ability to define or delete an alias related to a user catalog without any other security authority. You can still define or delete an alias if you have alter authority to the catalog, even if you do not have read authority to this facility class.

STGADMIN.IGG.DEFNVSAM.NOBCS
Controls the ability to define a non-VSAM data set with no BCS entry. Only a VVDS record (an NVR) for the system-managed non-VSAM data set is created.

STGADMIN.IGG.DEFNVSAM.NONVR
Controls the ability to define a non-VSAM data set with no VVDS entry (an NVR). Only a BCS entry for the system-managed non-VSAM data set is created.

STGADMIN.IGG.DELETE.NOSCRTCH
Controls the ability to delete the BCS entry for a system-managed data set without deleting the data set itself (for example, using DELETE NOSCRATCH). This controls functions which uncatalog data sets.

STGADMIN.IGG.DELGDG.FORCE
Controls the ability to use DELETE FORCE on a generation data group which contains a system-managed generation data set.

STGADMIN.IGG.DELNVR.NOBCSCHK
Controls the ability to delete the VVDS entry (the NVR) for an system-managed non-VSAM data set without checking the BCS entry and catalog name for the data set. If there is a BCS entry or if the catalog name contained in the NVR does not match the catalog provided in the request, the function is denied unless the user has authority to this profile.

STGADMIN.IGG.DIRCAT
Controls the ability to direct a catalog request to a specific catalog, bypassing the normal catalog search. A directed catalog request is one in which the catalog name is explicitly passed to the catalog in the CATALOG parameter of access method services commands.

In an SMS environment, all the catalog requests against system-managed data sets should be satisfied by the normal catalog search order. Directing the catalog request to a specific catalog requires authority to this profile with the exception of LISTCAT and EXPORT DISCONNECT requests.

STGADMIN.IGG.DLVVRNVR.NOCAT
Controls the ability to delete a VVR or NVR without an associated catalog. Users having RACF READ authority to the facility class need no other RACF authority to the master catalog to perform the DELETE VVR or DELETE NVR functions.

Caution: Restrict access to this facility class to users who understand the risk involved in deleting a VVR or NVR entry from a VVDS.

When a catalog is deleted for recovery purposes, or under certain failure conditions, an uncataloged VSAM data set or SMS nonVSAM data set can be left on the volume. The user can issue the DELETE VVR or DELETE NVR command to clean up the volume. In order to do this, the user needs RACF ALTER authority to the master catalog, and the user catalog must exist so that the catalog can be searched to verify that a BCS entry does not exist for the VVR or NVR. This is the usual situation when RACF ALTER authority to the catalog is needed. If the user catalog does not exist, the user must define an empty user catalog so that it can be searched.

The STGADMIN.IGG.DLVVRNVR.NOCAT facility class allows the use of DELETE VVR or DELETE NVR without an associated user catalog. It does not require RACF authority to the master catalog for these commands.

STGADMIN.IGG.LIBRARY
Controls the ability to DEFINE, DELETE, or ALTER library and volume entries in a tape library.

STGADMIN.IGWSHCDS.REPAIR
Controls the ability to use the AMS SHCDS command functions, which you can use to list outstanding SMSVSAM recovery and control that recovery.

See *z/OS DFSMSdss Storage Administration Guide* for more information on DFSMSdss functions.

Authority to Activate a Storage Management Subsystem Configuration

The FACILITY resource class profile STGADMIN.IGD.ACTIVATE.CONFIGURATION protects the ability to activate an SMS configuration. If you issue the ACTIVATE command from the Control Data Set Application Selection panel of ISMF, and either the Facility Resource Class is inactive or the named profile does not exist, the operator is queried to decide whether the ACTIVATE action should be allowed. You remain on the panel without the ability to provide additional input until action is taken from the operator console. If you have authority to activate an SMS configuration from ISMF, activation proceeds without confirmation from the operator console, and you continue with normal operations.

Storage Administrator Facility Class Profiles for DFSMSdss

Your installation can define one or more of the following special DFSMSdss storage administrator FACILITY class profiles:

STGADMIN.ADR.CONVERTV
Controls the ability to use the DFSMSdss CONVERTV command.

STGADMIN.ADR.COPY.BYPASSACS
Controls the ability to have the DFSMSdss COPY command bypass the ACS routines and use the original or user-specified storage class and management class.

STGADMIN.ADR.COPY.INCAT
Controls the ability to use the INCAT parameter on a DFSMSdss COPY command.

STGADMIN.ADR.COPY.PROCESS.SYS
Controls the ability to process system data sets with the DFSMSdss COPY command.

STGADMIN.ADR.DUMP.INCAT
Controls the ability to use the INCAT parameter on a DFSMSdss DUMP command.

STGADMIN.ADR.DUMP.PROCESS.SYS
Controls the ability to process system data sets with the DFSMSdss DUMP command.

STGADMIN.ADR.RELEASE.PROCESS.SYS
Controls the ability to process system data sets with the DFSMSdss RELEASE command.

STGADMIN.ADR.RESTORE.BYPASSACS
Controls the ability to have the DFSMSdss RESTORE command bypass the ACS routines and use the original or user-specified storage class and management class.

STGADMIN.ADR.STGADMIN.COMPRESS
READ access to this profile permits you to perform a COMPRESS without needing UPDATE access authority to the data sets being compressed.

STGADMIN.ADR.STGADMIN.COPY
READ access to this profile permits you to perform COPY data without having READ or UPDATE access authority to the data sets being copied. This includes the ability to read the catalog for the source data set and update the catalog for the target data set.

STGADMIN.ADR.STGADMIN.COPY.DELETE
READ access to this profile permits you to specify the DELETE parameter with the COPY command and the ADMINistrator parameter. With this, you can delete a data set without having to have ALTER access authority to the data set or ALTER access authority to the catalog for that data set. RACF can still be called in order for DFSMSdss to determine if erase-on-scratch processing for a given data set needs to be performed.

STGADMIN.ADR.STGADMIN.COPY.RENAME
READ access to this profile permits you to specify the RENAME or RENAMEU parameter with the COPY command and the ADMINistrator parameter. With this, you can copy the data set with a new name without having to have sufficient access authority to the new name for the target data set or to the catalog for the target data set.

STGADMIN.ADR.STGADMIN.DEFRAG
READ access to this profile permits you to perform a DEFRAG without having READ authority to the data sets that are moved. RACF can still be called in order for DFSMSdss to determine if erase-on-scratch processing for a given data set needs to be performed.

STGADMIN.ADR.STGADMIN.DUMP
READ access to this profile permits you to DUMP data without having READ access authority to the data sets being dumped. This includes the ability to read the catalog for the source data set.

STGADMIN.ADR.STGADMIN.DUMP.DELETE
READ access to this profile permits you to specify the DELETE parameter with the DUMP command and the ADMINistrator parameter. With this, you can delete a data set without having ALTER access authority to the data set or ALTER access authority to the catalog for that data set. RACF can still be called in order for DFSMSdss to determine if erase-on-scratch processing for a given data set needs to be performed.

STGADMIN.ADR.STGADMIN.PRINT
READ access to this profile permits you to print data without having READ access authority to the data being printed.

STGADMIN.ADR.STGADMIN.RELEASE
READ access to this profile permits you to RELEASE allocated but unused space without having UPDATE access authority to the data sets.

STGADMIN.ADR.STGADMIN.RESTORE
READ access to this profile permits you to RESTORE data without having READ, UPDATE, or ALTER access authority to the data being restored. This includes the ability to read the catalog for the source data set and update the catalog for the target data set.

STGADMIN.ADR.STGADMIN.RESTORE.RENAME
READ access to this profile permits you to specify the RENAME or RENAMEU parameter with the RESTORE command and the Administrator parameter. With this, you can restore the data set with a new name without having sufficient access authority to the new name for the target data set or to the catalog for the target data set.

See *z/OS DFSMSdss Storage Administration Guide* for more information on DFSMSdss functions.

FIELD Resource Class

The RACF FIELD resource class controls the ability of users to specify or update the following RACF profile fields:

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

How Authorization and Protection of Classes Differ

The *authorization* of SMS classes differs from the *protection* of SMS classes. For example, if you withhold authorization for a particular storage class, end users cannot specify that storage class in their JCL stream. End users specifying the storage class receive a JCL error if the ACS routine doesn't override the error.

Protection involves more than using an SMS class, it involves changing one. All end users might have READ access to the data set containing a particular storage class, but only a select few have UPDATE authority. Also, only people with UPDATE authority to the SCDS can update the SMS classes.

Authorizing the Storage and Management Classes

You can authorize the use of storage classes and management classes. Data classes do not require authorization.

Two of RACF's resource classes are: STORCLAS and MGMTCLAS. You authorize storage classes and management classes by defining them as RACF profiles to the general resource classes. The two resource classes, storage class and management class, are distinct from SMS storage class and management class.

Protecting the Control Data Sets

You need to protect three types of SMS data sets:

- Control data sets
  SCDS
  ACDS
  COMMDS
- ACS Routine Source Data Sets
  - Partitioned data sets whose members are source ACS routines
  - Sequential data sets that contain one source ACS routine.
- ACS Routine test library
  Partitioned data sets whose members are ACS test cases.

You can issue SECURITY from the command line of the ISMF Data Set Application to protect these data sets. You can give universal read access to ACDS, COMMDS, and SCDS to anyone. Anyone who is authorized to update an SCDS or an ACS routine source data set is also authorized to change any items within these data sets.

Authorizing for TSO and ISPF

To define an SMS data set under TSO, you do not need APF authorization, but it is recommended because APF authorization reduces path length.

For the DEFINE and IMPORT commands, you can establish this authorization by adding the DEFINE and IMPORT command processor names to the installation access list APFCTABL in CSECT IKJEFTE2, or the parameter AUTHCMD NAMES in SYS1.PARMLIB member IKJTSO00. For a call of access method services, you can establish this authorization by adding IDCAMS to the installation access list APFPTABL in CSECT IKJEFTE8 or the parameter AUTHPGM NAMES in SYS1.PARMLIB member IKJTSO00. For more information about specifying authorized commands and programs under TSO, see *z/OS TSO/E Customization*.

To define or import an SMS data set while under option 6 of ISPF/PDF, you must include the DEFINE and IMPORT processor names in the command table in ISPTCM. See *z/OS ISPF Planning and Customizing* for more information.

Storage Administrator Authorization

Storage administrators should receive the following authority from the security administrator:

- CLAUTH to STORCLAS and MGMTCLAS resource classes
- Authority to activate an SMS configuration
- Authority to perform exceptional catalog and DFSMSdss operations on system-managed data sets
- Authority to all ISMF applications, dialogs, line operators, and commands
- Update authority to the RESOWNER field in the DATASET profiles
- Update authority to the DATAAPPL, MGMTCLAS, STORCLAS, and DATACLAS, fields in the USER/GROUP profiles
- Authority to alter SMS constructs.

It might be preferable to have a group profile with authorities mentioned above for all storage administrators.

Chapter 14. Administering VSAM Record-Level Sharing

VSAM record-level sharing (RLS) is a data set access mode that allows multiple address spaces, CICS application owning regions (AORs) on multiple MVS systems, and jobs to access data at the same time. With VSAM RLS, multiple CICS systems can directly access a shared VSAM data set, eliminating the need for function shipping between application owning regions (AORs) and file owning regions (FORs). CICS provides the logging, commit, and rollback functions for VSAM recoverable files; VSAM provides record-level serialization and cross-system caching. CICS, not VSAM, provides the recoverable files function.

This chapter assumes you are familiar with the concepts of VSAM RLS and how it uses the coupling facility (CF) cache and lock structures. It describes the following tasks to enable and maintain VSAM RLS:

- "Preparing for VSAM Record-Level Sharing"
- "Defining CF Cache Structures in the SMS Base Configuration" on page 247
- "Defining Storage Classes for VSAM RLS" on page 250
- "Activating VSAM RLS" on page 252
- "Monitoring the Coupling Facility for VSAM RLS" on page 253
- "Recovering VSAM RLS Processing" on page 259
- "Falling Back from VSAM RLS Processing" on page 261

See *z/OS DFSMS: Using Data Sets* and *CICS Transaction Server for OS/390 Migration Guide* for more information on VSAM RLS. Go to `http://www.ibm.com/s390/pso/psotool` for information on the online Parallel Sysplex Configuration Assistant.

Preparing for VSAM Record-Level Sharing

Planning for and installing VSAM RLS requires coordination with system hardware and software groups. This section describes the tasks that the storage administrator must do with the MVS systems programmer and the CICS database administrator to prepare to use VSAM RLS:

- "Determining Hardware Requirements"
- "Understanding the Product Environment for VSAM RLS" on page 236
- "Determining Applications That Can Use VSAM RLS" on page 236
- "Ensuring Same Systems Connectivity" on page 237
- "Planning for Availability" on page 238
- "Defining Sharing Control Data Sets" on page 239
- "Defining CF Cache Structures" on page 241
- "Defining the CF Lock Structure" on page 242
- "Modifying the SYS1.PARMLIB IGDSMSxx Member" on page 247
- "Establishing Authorization for VSAM RLS" on page 247

Determining Hardware Requirements

Recommendation: Use multiple CFs with global connectivity. This ensures maximum availability, simplifies systems management and allows for nondisruptive lock transfer in the event of a CF outage.

You must have at least one CF connected to all systems capable of VSAM RLS within the Parallel Sysplex (for lock structure duplexing, you must have at least two CFs). For multiple CFs, select one facility with global connectivity to contain the master lock structure. For maximum availability, use a second CF with global connectivity. In the event a CF becomes unavailable, the SMSVSAM address space

© Copyright IBM Corp. 1986, 2002

can transfer its in-storage duplexed copy of the locks to the other available CF without causing any application disruption.

You can attach CFs which do not contain the master lock structure to a subset of the systems. A CF cache structure that is referenced by a storage class cache set must have at least the same connectivity as the storage groups to which the storage class maps.

The following hardware requirements are necessary to utilize system-managed lock structure duplexing:

- CFLEVEL = 10 CF architecture support
- CFCC licensed internal code implementation
- CF-to-CF peer links
- A duplexed pair of primary and secondary VSAM RLS lock structure instances
- Message architecture LIC support for each VSAM RLS system image that has a connector to a duplexed lock structure

Depending upon your existing configuration, you might also need additional CF processor, storage, and link capacities to utilize system-managed duplexing.

Understanding the Product Environment for VSAM RLS

VSAM RLS processing involves support from multiple products, such as CICS, CICSVR, and DFSMS. See *CICS Recovery and Restart Guide* for information on CICS and CICSVR support for VSAM RLS. See *z/OS DFSMS Migration* for information on coexistence of different releases supporting VSAM RLS.

Determining Applications That Can Use VSAM RLS

Applications using VSAM RLS benefit from increased data availability inherent in a shared environment that has read/write integrity and record-level locking (as opposed to control interval (CI) locking).

Your applications should fall into one of the following categories:

VSAM RLS-tolerant application
The application runs correctly in a multi-update environment when RLS is specified in the JCL or when MACRF=RLS is specified on the ACB. The RLS JCL parameter allows batch read programs to use RLS without requiring a recompile of the program. For batch update programs running against nonrecoverable VSAM RLS spheres, it might be possible to modify the allocation from DISP=OLD to DISP=SHR.

VSAM RLS-exploiting application
The application recognizes when a VSAM data set can be shared at the record level and uses VSAM RLS functions to access the data. CICS applications are exploiting applications.

VSAM RLS-intolerant application
The application uses facilities not supported by VSAM RLS, accesses VSAM internal data structures, or is incompatible with the functions of VSAM RLS. For example, the application might perform CI access against the data set.

However, since VSAM RLS can be used by CICS and non-CICS applications, the CICS recoverable files function provides transactional recovery for applications, VSAM RLS is expected to be used primarily by CICS applications. Transactional recovery isolates the changes made by each sharing application. CICS creates a

backout log record for each change made to a recoverable file, and VSAM RLS obtains and holds a lock on each changed record. The lock remains held until the transaction ends. If a transaction fails, CICS backs out all changes made by the application to recoverable files, thus isolating the other sharing applications from the failure.

For recoverable data set, in addition to the data sharing across CICS applications, VSAM RLS enables read-with-integrity sharing by batch jobs. Batch jobs can share recoverable files while they are being modified by CICS applications. This is possible because VSAM provides the record locking and buffer coherency functions across CICS and batch. Since VSAM RLS does not provide the transactional recovery function for batch jobs, it does not allow a batch job to open a recoverable data set for output.

VSAM RLS permits read and write sharing of nonrecoverable data sets across CICS and batch jobs. Transactional recovery does not apply to nonrecoverable data sets. While VSAM RLS permits sharing, the jobs must be carefully designed to achieve correct results in a read/write data sharing environment, since they do not have the isolation provided by transactional recovery.

Ensuring Same Systems Connectivity

You must ensure same systems connectivity for CF cache structures, lock structure, and storage groups. This is to ensure that jobs running in the Parallel Sysplex have access to data in both the CFs and storage groups. The lock structure must have global connectivity to all systems in the Parallel Sysplex.

Figure 111 shows how connectivity among systems and storage groups matches connectivity among systems and coupling facilities.

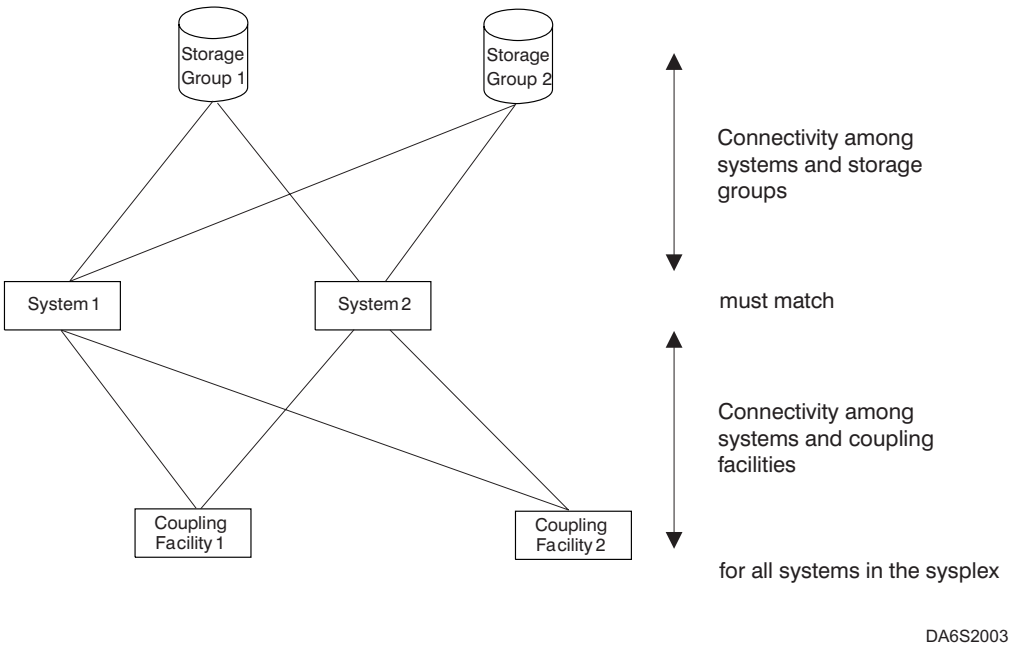


Figure 111. Example of Global Connectivity in a Parallel Sysplex

Figure 112 on page 238 illustrates the problems that can occur if you do not have global connectivity across the Parallel Sysplex:

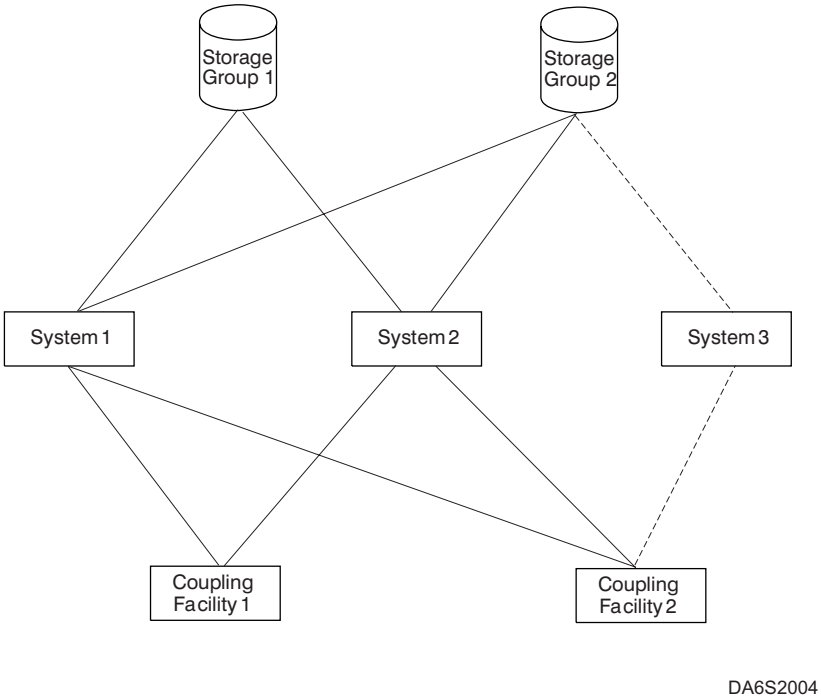


Figure 112. Example of Insufficient Connectivity in a Parallel Sysplex

In Figure 112, System 3 is connected to both Storage Group 2 and to Coupling Facility 2. This meets the minimum connectivity requirements but can still cause problems. Since Storage Group 2 data can be placed in Coupling Facility 1, jobs in System 3 can fail if they attempt to access data sets in Storage Group 2 that have been assigned to Coupling Facility 1. You can avoid such a scenario by ensuring that, for all systems in the Parallel Sysplex, connectivity among systems and storage groups matches connectivity among systems and coupling facilities.

Planning for Availability

In order for VSAM RLS processing to take place, you must ensure the following:

- Run all systems performing RLS as a Parallel Sysplex
- Define and activate at least two sharing control data sets (SHCDS), and one spare SHCDS for recovery purposes
- Define CF cache and lock structures to MVS, using the CF resource manager (CFRM) policy. Define CF cache structures in the SMS base configuration.
- Associate CF cache set names with storage class definitions, and write ACS routines to associate storage class definitions that map to CF cache structures with data sets.
- Change the attributes for a data set to specify whether the data set is to be recoverable or nonrecoverable. Specify LOG(NONE) if the data set is nonrecoverable; specify LOG(UNDO) or LOG(ALL) if the data set is recoverable.

Related Reading:

- For more information about using the CFRM policy to define CF structures, see *z/OS MVS Setting Up a Sysplex*.
- For more information about specifying recoverable and nonrecoverable data set attributes, see *z/OS DFSMS: Using Data Sets*.

Defining Sharing Control Data Sets

Sharing control is a key element in maintaining data integrity in a shared environment. Because persistent record locks are maintained in the CF, several new classes of failure could occur, such as a Parallel Sysplex, system, or SMSVSAM address space restart, or a CF lock structure failure. The sharing control data set (SHCDS) is designed to contain the information required for DFSMS to continue processing with a minimum of unavailable data and no corruption of data when failures occur.

The SHCDS acts as a log for sharing support. It is a logically-partitioned linear data set that can be defined with secondary extents, though all extents for each data set must be on the same volume. An SHCDS contains the following information:

- The name of the CF lock structure in use
- A list of subsystems and their status
- A list of open data sets using the CF
- A list of data sets with unbound locks
- A list of data sets in permit non-RLS state

Sharing control is critical to maintaining data integrity in the event of the failures such as Parallel Sysplex, system, or SMSVSAM address space restart, or of the CF lock structure. You should always have at least two active and one spare SHCDSs. Place these data sets for maximum availability. If necessary, you can have up to five active SHCDSs and five spare SHCDSs.

Recommendation: Because the contents of these data sets are highly dynamic, do not use backup and restore functions, since these could cause the loss of VSAM RLS data set recovery information.

Consider the following as you allocate and maintain your SHCDSs:

- Allocate SHCDSs so that the number of active and spare data sets ensures the data is always duplexed. At a minimum, define and activate two SHCDSs and at least one spare SHCDS for recovery purposes. You should ensure that there are enough spare SHCDSs, since these are used when I/O errors occur on the active SHCDSs.
- Place the SHCDSs on volumes with global connectivity. VSAM RLS processing is only available on those systems that currently have access to the active SHCDS. The share options for SHCDSs must be set to (3,3) so that each system in the Parallel Sysplex can properly share the data sets.
- Place your SHCDSs in such a way as to maximize availability in the event of the loss of a volume. Use storage classes defined with the guaranteed space attribute. Avoid placing SHCDSs on volumes for which there might be extensive volume reserve activity.
- Ensure that the space allocation for active and spare SHCDSs is the same. Ensure data sets have enough space for growth.
- Use the VARY SMS,SHCDS command to activate and maintain your SHCDSs.

You must use the following naming convention when defining your SHCDSs:

SYS1.DFPSHCDS.*qualifier***.V***volser*

where:

qualifier
: is a 1 to 8 character qualifier.

volser is the volume serial number. The **V** prefix allows you to specify numeric volume serial numbers.

Use the following formula to calculate the size of your SHCDS:

Space = (16 + (*number_of_systems* x (16 + *number_of_OPENs*/10))) kilobytes

where:

Space
is the space required for the SHCDS.

number_of_systems
is the number of systems.

number_of_OPENs
is the number of concurrent OPENs to the CF that you expect.

For example, if you have 3 systems and expect to have 2000 concurrent OPEN requests, your SHCDS requires 664 KB of space.

You can create an SHCDS using JCL, or using access method services or TSO subcommands. When you use the access method services DEFINE command to create an SHCDS, specify SHAREOPTIONS(3,3) to ensure that the SHCDS can be written to and read from any system.

When you use JCL, the default for share options is (1,3). If you use JCL to define the SHCDSs, use a data class defined with a cross-region share options value of 3 and a cross-system share options value of 3.

Figure 113 shows how to create an SHCDS using JCL:

```
//PRISHCDS DD DSN=SYS1.DFPSHCDS.PRIMARY.VSMS001,SPACE=(1,(10,10)),
//            RECORG=LS,STORCLAS=GSPACE,VOL=SER=SMS001,AVGREC=M,
//            DISP=(NEW,CATLG)
//SECSHCDS DD DSN=SYS1.DFPSHCDS.SECONDRY.VSYS002,SPACE=(1,(10,10)),
//            RECORG=LS,VOL=SER=SYS002,AVGREC=M,
//            DISP=(NEW,CATLG)
//SPRSHCDS DD DSN=SYS1.DFPSHCDS.SPARE.VSMS003,SPACE=(1,(10,10)),
//            RECORG=LS,STORCLAS=GSPACE,VOL=SER=SMS003,AVGREC=M,
//            DISP=(NEW,CATLG)
```

Figure 113. JCL Example for Creating Sharing Control Data Sets

Once the SHCDS data sets have been created, you make them available for use by using the `VARY SMS, SHCDS (qualifier.Vvolser)`, NEW command. The `(qualifier.Vvolser)` is used, not the SYS1.DFPSHCDS command. This command only needs to be entered on one system in the Parallel Sysplex. The SMSVSAM address space on the system where the command is entered communicates the name of the data set to the other SMSVSAM address spaces in the Parallel Sysplex. Each SMSVSAM address space recatalogs the data set, if it has not already been cataloged, so that you do not need to manually catalog the data set in order for it to be used. Those data sets added for use are saved, and can be accessed when an SMSVSAM address space initializes them or is restarted, or though subsequent IPLs.

Defining CF Cache Structures

CF cache structures must be defined to MVS and also in the SMS base configuration. CF cache structures provide a level of storage hierarchy between local memory and DASD cache. They are also used as a system buffer pool for VSAM RLS data when that data is modified on other systems. Each CF cache structure is contained in a single CF. You might have multiple CFs and multiple CF cache structures.

Several factors determine the number and size of your CF cache structures:

- Number of available CFs
- Amount of space available in each CF
- Amount of data to be accessed through each CF
- Continuous availability requirements for CF reconfiguration
- Performance requirements for various applications

You use CFRM policy definitions to specify an initial and maximum size for each CF cache structure. DFSMS uses the initial structure size you specify in the policy each time it connects to a CF cache structure.

You can assign one or more CF cache structures to each cache set associated with a storage class. Having multiple cache sets allows you to provide different performance attributes for data sets with differing performance requirements. When more than one CF cache structure is assigned to a cache set, data sets are assigned to each CF cache structure in an effort to balance the load.

Determining CF Cache Structure Size

A CF cache structure must be at least large enough to hold all of the MVS information required to describe a structure of maximum size. To help you achieve the best possible performance with VSAM RLS buffering, the sum total of all the CF cache structure sizes you define (the CF cache) should ideally be the sum total of the local VSAM local shared resources (LSR) buffer pool sizes. The size of the local VSAM LSR buffer pool is the sum of LSR pool size and, if used, the corresponding Hiperspace™ pool size. You can run VSAM RLS with less CF cache storage than this, but the CF cache *must* be large enough for the CF cache directories to contain an entry for each of the VSAM RLS local buffers across all instances of the SMSVSAM server. If the CF cache cannot contain the directory entries describing the local buffers, then the VSAM RLS local buffers are falsely invalidated and must be refreshed. To minimize this, the minimum CF cache structure size should never be less than 1/10 the size of the local buffer pool.

For example, the following CICS FOR configuration shows the sum total of the local VSAM RLS buffer pool size prior to migrating to VSAM RLS.

File Owning Region	LSR pool size	Hiperspace pool size	Sum Total
FOR_1	20 MB	30 MB	50 MB
FOR_2	40 MB	no pool	40 MB
FOR_3	30 MB	50 MB	80 MB
			170 MB

When migrating this configuration to VSAM RLS, the CF cache you define should ideally be at least 170MB. In this way, cross-invalidated local RLS buffers can be refreshed from the CF cache structures.

Performance should improve when the CF cache is larger than the sum of the local VSAM LRS buffer pool sizes. When the CF cache is smaller, performance depends upon the dynamics of the data references among the systems involved. In some cases, you might want to consider increasing the size of very small CF caches (2 MB - 10 MB).

In those situations where you can determine that data previously treated as a nonshared resource (NSR) is no longer to be treated as such, you should also include NSR buffer sizes in the total local buffer pool size.

See *CICS Transaction Server for OS/390 Migration Guide* for more information on calculating a cache structure size.

Using the RLS_MAX_POOL_SIZE Parameter to Limit Local Buffer Pool Size

You can use the RLS_MAX_POOL_SIZE parameter of the IGDSMSxx parmlib member to limit the maximum size of the local buffer pool(s). The value you specify can be either larger or smaller than the default maximum pool size of 100 MB, but it must be supported with the available real and expanded storage. Although VSAM RLS can in some cases ascertain how much buffer space is supported before paging begins to occur, specifying an RLS_MAX_POOL_SIZE value ensures that the local buffer pool does not grow beyond the value you specify.

Tip: In some instances, the local buffer pool might *temporarily* grow larger than the RLS_MAX_POOL_SIZE value. Setting a RLS_MAX_POOL_SIZE value that is too low might result in unnecessarily degrading the local hit rate.

Recommendation: Initially, set the RLS_MAX_POOL_SIZE value to 50% more than the sum of the local buffers on a single system. The following table illustrates this, assuming that each CICS FOR is on a separate system:

File Owning Region	LSR pool size	Hiperspace pool size	Sum Total	RLS_MAX_POOL_SIZE
FOR_1	20 MB	30 MB	50 MB	75 MB
FOR_2	40 MB	no pool	40 MB	60 MB
FOR_3	30 MB	50 MB	80 MB	120 MB

You can use the information from the SMF type 42 record with subtype 19 to evaluate the local buffer hit rates for each of the individual systems. Use this information together with the local system paging rates to help you make additional tuning adjustments to the RLS_MAX_POOL_SIZE parameter.

Defining the CF Lock Structure

Requirements: To use VSAM RLS, you must define a single, master CF lock structure. For system-managed duplexing, you must define two CF lock structures.

For maximum availability, define nonvolatile lock structures. CF lock structures enforce the protocol restrictions for VSAM RLS data sets and maintain the record-level locks and other DFSMSdfp serialization. Ensure that the CF lock structures have universal connectivity, so that they are accessible from all systems in the Parallel Sysplex that support VSAM RLS. For system-managed duplexing, define the lock structure as duplexed (ENABLE or ALLOW) in the CFRM policy.

For more information about system-managed lock structure duplexing, see “Recovering the CF Lock Structure” on page 259.

The CF lock structure is named IGWLOCK00. Use the XES coupling definition process to define it. To estimate its size requirements in megabytes, use the following formula (a megabyte is 1048576 bytes in this case):

```
10M * number_of_systems * lock_entry_size
```

where:

number_of_systems
: is the number of systems in the Parallel Sysplex

lock_entry_size
: is the size of each lock entry. This value depends on the MAXSYSTEM value that is specified to the IXCL1DSU Couple Data Set format utility.

  Use the following table to determine the actual lock entry size for the different MAXSYSTEM setting values:

Table 20. Effect of MAXSYSTEM Value on Lock Table Entry Size

MAXSYSTEM Value	Lock Entry Size
7 or less	2 bytes
>= 8 and < 24	4 bytes
>= 24 and <= 32	8 bytes

Table 21 shows some sample lock allocation estimates:

Table 21. CF Lock Structure Sizing Examples

MAXSYSTEM Value	Number of systems	Total Lock Size
<= 7	2	40MB
	4	80MB
<= 23	8	320 MB
<default> =8	2	80 MB
	4	160 MB
	8	320 MB
32	2	160 MB
	4	320 MB
	8	320 MB

These lock size estimates include the memory requirements for both the lock table and the record-lock memory. Use these estimates as rough initial values to help you attain a locking structure with a desired false contention target of approximately one-half of 1% or less. Contact your marketing representative for help in arriving at an initial estimate that more closely matches your specific configuration.

Considerations for Retained Locks and Record Table Full Conditions

The CF lock structure includes two parts:

- a lock table, used to determine whether there is R/W interest among systems on a particular resource
- record table space to keep track of information for retained locks and spheres which have been processed by VSAM RLS

If a commit protocol application such as CICS fails, the locks protecting updates against recoverable spheres are remembered in the record table space until CICS

performs the required backouts. Also, update locks associated with indoubt transactions are remembered until the indoubts are resolved.

When used record table space reaches 80% or greater, informational message IGW326I is issued. A shortage of record table space can occur for the following reasons:

- The size of the lock structure is too small for normal system operation.
- A CICS system has failed and cannot be successfully restarted to run its backouts. As a consequence, record table space cannot be freed up.
- Outstanding indoubt transactions exist. CICS provides commands to display indoubt transactions and resolve them.
- Transactions are in backout failed state. This means that the backouts could not complete and free record table space.

Since it might be difficult to remedy these situations quickly, you should respond to a record table shortage by modifying the CFRM policy to increase the size of the lock structure, in order to provide additional record table space and increase the space available for retained locks. Then, activate that policy and rebuild the lock structure via operator command. The operator command can be used to change the size of the lock structure.

You can increase record table space by carefully selecting the total lock memory allocation quantity. Record table space is guaranteed to never be less than one-half of the total lock space allocated to RLS, but it can be more if the total size is not a power of two. A lock space allocation of 32 MB causes 16 MB to be allocated to the lock table and 16 MB to be allocated to the record lock table. An allocation of 63 MB, on the other hand, causes 16MB to be allocated to the lock table, and all of the remaining memory (for example. 47 MB) is then allocated to the record table space. Thus, selecting a total lock size that is not a power of two value is a means of causing the record space to grow without necessarily increasing the table size.

When you attempt to change the size of the lock structure using either the rebuild or the alter function and the record table size is greater than 50%, VSAM RLS will determine whether the new table has enough space for all existing record table entries, plus enough empty space for future locking processes (120%). If the answer is no, message IGW322I is displayed. A rebuild request is rejected; an alter request proceeds.

If the record table becomes full, VSAM requests which require that a lock be recorded receive RPL feedback, and CICS backs out the transaction. A record table full condition means that transactions cannot complete successfully, and the size of the lock structure *must* be increased.

Avoiding False Contention

VSAM RLS assigns locked resources to an entry value in the lock table, and uses this entry value to quickly check whether a resource is already locked. If the lock structure (and thus the lock table) is too small, many locks can be represented by a single value, making ″false″ lock contention possible. False lock contention occurs when two different locks on different resources attempt to use the same lock entry. The second lock requester is suspended until VSAM RLS determines that there is no real lock contention on the resource. False contention can be a problem for workloads with heavy R/W interests among systems.

To avoid false contention, you need to consider the size of the lock table. The lock table size is determined by the total size of the lock structure. When you define the

size of the total lock structure, you should specify a value that is a power of two in order to maximize the lock table size: the lock table comprises 50% of the total space, and the record lock space the remaining 50%. Any memory in excess of a power of two value is allocated to the record lock space exclusively until the next power of two value is reached. At that time, the lock table space is doubled, and the two allocations are once again of equivalent sizes.

VSAM RLS uses the MAXSYSTEM value from the Couple Data Sets format utility to determine the size of each lock entry. Since smaller lock entry sizes imply a larger number of locks for the same memory allocation, it is very important to select an appropriate MAXSYSTEM value. The MAXSYSTEM value represents the maximum number of systems that can be connected into the Parallel Sysplex. There is a lock table memory penalty when the MAXSYSTEM setting exceeds 7, and another penalty when it exceeds 23. Consequently, from a false contention point of view you want to select a MAXSYSTEM value that does not exceed 7 or 23.

To increase a MAXSYSTEM value that has already been specified, you must first format larger CDSs and switch them into use dynamically. Then, either manually rebuild the CF lock structure using the SETXCF START command or start system-managed duplexing rebuild using the SETXCF START,REBUILD,DUPLEX command.

To decrease a MAXSYSTEM value that has already been specified, you must shut down all of the VSAM RLS address spaces, and manually delete both the persistent connections and the lock structure. Then you must restart the VSAM RLS address spaces. If you decrease the MAXSYSTEM value without first shutting down, the decrease has no effect; the Parallel Sysplex continues to run using the old MAXSYSTEM value.

Monitoring for False Contention: You can determine the amount of false contention by using either the resource measurement facility (RMF) or the DISPLAY SMS,CFLS command.

How Much Contention is Acceptable: For the best performance, you want to achieve the least possible amount of global lock contention, both real and false. The amount of real lock contention is application-dependent; it depends on record access patterns. False lock contention is almost entirely determined by the size of the lock table, with a larger lock table having less false lock contention than a smaller one. A good goal is to have total (real and false) global lock contention of less than one percent. The false contention component of the total global lock contention should be less than one-half of one percent, and ideally, should be substantially less than this.

Reducing False Contention: If false contention becomes a problem, try the following:

- Reduce the amount of real lock contention in your applications, if possible
- Specify a larger size for the lock structure and manually rebuild it
- Ensure the MAXSYSTEM parameter of the Couple Data Set utility is not too large for the number of members in your Parallel Sysplex

  A MAXSYSTEM value of 7 or less allows you twice as many lock entries as a MAXSYSTEM value of 8, as shown in Table 20 on page 243.

Adjusting the Lock Structure Size

Once you select an initial lock structure size and the Parallel Sysplex has been running with that size for some time, you should monitor the percentage of false contentions. You can then use this percentage to help you select an even more appropriate lock structure size.

Use the following formula to determine the estimated lock structure size based on a false contention percentage:

```
Minimum Lock Structure Size = F * M / T
```

where:

F is the measured false contention percentage, expressed as a percentage (for example, a rate of 0.02 = 2%, so F = 2)

M is the current lock structure allocation size

T is the target false contention target percentage, expressed as a percentage (for example, a rate of 0.005 = 0.5%, so T = 0.5)

A Lock Structure Sizing Example

Suppose a Parallel Sysplex has a MAXSYSTEM setting of 3, with two systems currently connected to the CF. You could use the initial sizing formula to estimate its initial size, as shown below:

```
Initial Lock Structure Size = 10M * 2 * 2
```

where:
number_of_systems = 2
lock_entry_size = 2 bytes

This yields an initial lock structure size of 40MB. However, to maximize the lock table space itself, you should size it with a number that is a power of two. In this case, the initial total lock size could be set at either 32 or 64 MB.

For the purpose of this example, we select 32 MB as the initial lock structure size. We then run for a while before determining the rate of false contentions. Assuming that this false contention rate is 1.5%, and that we have a target false contention rate of 0.5%, we can then use the following formula to modify the lock structure size:

```
Minimum Lock Structure Size = 1.5 * 32MB / 0.5
```

where:
1.5 is the measured false contention rate of 1.5%
32 MB is the specified lock structure size
0.5 is the target false contention rate of 0.5%

The adjusted size should now be 96 MB. However, since this needs to be expressed as a power of two, unless we actually select a value of at least 128 MB, it is unlikely that we will find that the false contention rate meets or exceeds our target of 0.5%.

In this example, the Parallel Sysplex would run better with a larger lock structure size than initially allocated. On the other hand, had the monitored false contention rate been less than an initial target value of 0.5%, it would not mean that we were wasting CF storage by having it allocated to the lock structure. In fact, although a false contention rate of 0.1% is not feasible in many cases, it is still ideal, assuming that it can be achieved with a reasonable amount of CF memory.

Modifying the SYS1.PARMLIB IGDSMSxx Member

The IGDSMSxx parmlib member includes the following parameters to support the CF and VSAM RLS processing:

- CF_TIME, which aligns creation of all the CF-related SMF type 42 records with subtypes 15, 16, 17, 18 and 19
- DEADLOCK_DETECTION, which specifies the interval for detecting deadlocks between systems
- RLSINIT, which specifies whether the SMSVSAM address space is started as part of system initialization or the V SMS,SMSVSAM,ACTIVE command
- RLS_MaxCfFeatureLevel, which specifies the method VSAM RLS caching uses to determine the size of the data that is placed in the CF cache structure.
- RLS_MAX_POOL_SIZE, which specifies the maximum size of the SMSVSAM local buffer pool
- SMF_TIME, which aligns the SMF type 42 records for DFSMS (with subtypes 1, 2, 15, 16, 17, 18 and 19) to the SMF_TIME interval

You can modify these parameters at any time during VSAM RLS processing. See "Changing IGDSMSxx Parameters to Support the Coupling Facility" on page 256 for information on changing these parameters.

Establishing Authorization for VSAM RLS

You want to establish the following authorization to restrict access to certain VSAM RLS capabilities:

- SMSVSAM should be authorized to update SYS1.DFPSHCDS.* data sets. If you protect SYS1.* data sets be sure SMSVSAM is able to access SYS1.DFPSHCDS.* for update.
- To use the access method services SHCDS command, you must be authorized to the facility class STGADMIN.IGWSHCDS.REPAIR. The SHCDS command is used to list SMSVSAM recovery associated with subsystems and spheres, and to control that recovery.
- Only those users who actually need the capability, such as CICS subsystems, should have access to register a subsystem name to SMSVSAM. Use the RACF subsystem name class to restrict this access. For more information, see *CICS Transaction Server for OS/390 Release Guide*.

Defining CF Cache Structures in the SMS Base Configuration

In order for DFSMSdfp to use the CF for VSAM RLS, after you define one or more CF cache structures to MVS, you must also add them in the SMS base configuration.

To add CF cache structures to the base configuration, you associate them with a cache set name. This cache set name is also specified in one or more storage class definitions. When a storage class associated with a data set contains a cache set name, the data set becomes eligible for VSAM record-level sharing and can be placed in one of the CF cache structures associated with the cache set. The system selects the best cache structure in which to place the data set.

Related Reading: See "Defining CF Cache Structures" on page 241 and "Defining the CF Lock Structure" on page 242 for more information on using CFRM policies to define CF cache and lock structures.

To define CF cache structures to DFSMS:

1. Select option 8 from the ISMF Primary Option Menu for Storage Administrators, to invoke the Control Data Set (CDS) Application Selection panel on ISMF.

   Figure 114 shows the CDS Application Selection panel:

```
  Panel  Utilities  Help
-------------------------------------------------------------------------------
                          CDS APPLICATION SELECTION
Command ===>

To Perform Control Data Set Operations, Specify:
  CDS Name . . 'USER6.TESTCDS'
                            (1 to 44 Character Data Set Name or 'Active')

Select one of the following Options:

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

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

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

Figure 114. CDS (Control Data Set) Application Selection for VSAM RLS

2. Specify the name of the SCDS that is to contain the base configuration for VSAM RLS in the CDS NAME field.

   When this panel is displayed, the CDS NAME field is primed with the last used SCDS name. This name might differ from the SCDS name of your base configuration for VSAM RLS.

3. Select option 7, CACHE UPDATE, and press Enter. ISMF displays the CF Cache Set Update panel, shown in Figure 115 on page 249.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
DGTDBSA3                      CF CACHE SET UPDATE                 Page 1 of 1
Command ===>

SCDS Name  : USER6.TESTCDS
Define/Alter/Delete CF Cache Sets:       ( 000  Cache Sets Currently Defined )

 Cache Set                      CF Cache Structure Names
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
 ________  ________________ ________________ ________________ ________________
           ________________ ________________ ________________ ________________
More CF Cache Sets to Add?  . . N  (Y/N)
Use ENTER to Perform Validation; Use UP/DOWN Command to View other Pages;
Use HELP Command for Help; Use END Command to Save and Exit.
```

Figure 115. Coupling Facility Cache Set Definition for VSAM RLS

4. Define your CF cache set(s).

   The CF Cache Set Update panel allows you to define up to 256 CF cache sets. Each CF cache set can have up to eight CF cache structure names assigned to it. Use the cache structure names you defined in the MVS CFRM policies.

Figure 116 on page 250 shows how CF cache structures can be defined in multiple cache sets, and also shows how data sets associated with a storage class definition containing a cache set name can be placed in multiple cache structures.

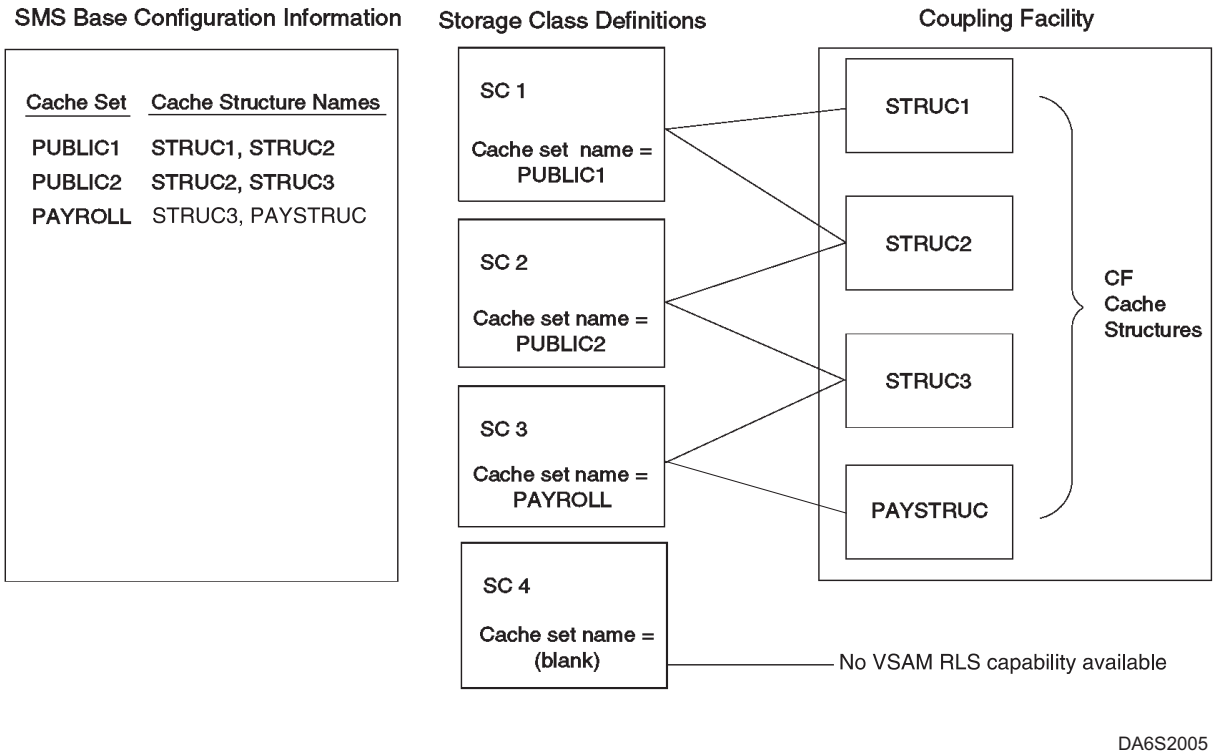


Figure 116. Example of CF Cache Structure Definition in Base Configuration

Defining Storage Classes for VSAM RLS

This section describes how to assign the cache set names defined in the base configuration to a storage class, so that data sets associated with that storage class can be eligible for VSAM RLS and use CF cache structures. It also describes how to indicate the relative importance of the data associated with the storage class.

Requirement for JES3 users: In a JES3 environment, be careful to define cache set names only in those SMS storage classes that are used by data sets opened for VSAM RLS processing. When you define a cache set name in a storage class, any job accessing a data set associated with that storage class is scheduled on a VSAM RLS-capable system. If all storage classes have cache set names defined for them, then all jobs accessing SMS-managed data sets are scheduled to VSAM-RLS-capable systems. This could cause a workload imbalance between those systems and down-level systems.

To assign the CF cache sets defined in the base configuration to storage classes, follow these steps:

1. Select option 5, STORAGE CLASS, from the ISMF Primary Option Menu for Storage Administrators. ISMF displays the Storage Class Application Selection panel, shown in Figure 117 on page 251:

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

To perform Storage Class Operations, Specify:
  CDS Name  . . . . . . . 'USER11.SCDS'
                              (1 to 44 character data set name or 'Active' )
  Storage Class Name  . . *         (For Storage Class List, fully or
                                    partially specified or * for all)
Select one of the following options  :
  3  1. List          - Generate a list of Storage Classes
     2. Display       - Display a Storage Class
     3. Define        - Define a Storage Class
     4. Alter         - Alter a Storage Class
     5. Cache Display - Display Storage Classes/Cache Sets
If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

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

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

Figure 117. Defining a Storage Class for VSAM RLS

2. Specify the name of the SCDS you defined for VSAM RLS in the CDS NAME field.
3. Select option 3, DEFINE, and press Enter. ISMF displays the Storage Class Define panel.
4. Press the DOWN key to view the second Storage Class Define panel, shown in Figure 118:

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

SCDS Name . . . . . : IBMUSER.TESTCDS
Storage Class Name  : STOR1

To  Storage Class, Specify:

  CF Cache Set Name . . . . . CSI         (up to 8 chars or blank)
  CF Direct Weight  . . . . . 6           (1 to 11 or blank)
  CF Sequential Weight  . . . 6           (1 to 11 or blank)









Use ENTER to Perform Verification; Use UP Command to View previous Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 118. Defining Storage Class Attributes for VSAM RLS

5. Enter the name of the CF cache set you defined in the base configuration.

6. Specify a weight attribute for the data in the CF Direct Weight or the CF Sequential Weight fields to indicate the data's relative importance. The default is a weight value of 6.

   Note: DFSMS only supports the default value. Regardless of what you specify, all data eligible for VSAM record-level sharing is assigned a weight value of 6.

Defining VSAM RLS Attributes in Data Classes

You can define data classes specifically for data sets eligible for VSAM RLS. When you have data classes with VSAM RLS parameters, you do not need to change current AMS DEFINE statements and job streams.

Recommendation: To avoid having too many data classes, you should specify the JCL LGSTREAM ID keyword for SMS VSAM data sets defined by JCL instead of the Logstream ID attribute in the data class. See "Defining Data Class Attributes" on page 112 for more information.

Activating VSAM RLS

The SMSVSAM address space always starts at IPL time, providing RLSINIT(YES) is specified in the IGDSMSxx parmlib member. VSAM RLS processing is available once certain minimum requirements are met. This section describes those requirements.

Enabling VSAM RLS Processing

VSAM RLS processing is available once the following requirements are met:

- All systems are running as a Parallel Sysplex.
- At least two SHCDSs, and one spare SHCDS have been activated.
- At least one existing CF cache structure is defined to MVS and to the SMS base configuration.

  If a subset of the cache structures is not available, some data sets might not be accessible.
- The CF lock structure IGWLOCK00 is available.
- The SMS address space is started.

If a null configuration is active, only existing data sets already assigned to a CF cache structure can be processed.

If VSAM RLS processing is not enabled, all attempts to open a data set where VSAM RLS is specified on the ACB or the JCL fail.

Under the following conditions, VSAM RLS access from a system is *not* available:

- No SHCDSs are available.
- The CF lock structure, IGWLOCK00, is not available.
- The SMSVSAM address space has failed and a response to message IGW418D is pending.
- The SMSVSAM address space has failed, and manual restart is in effect (the response to message IGW418D was **C**).
- None of the CF cache structures defined in the SMS configuration are available. If a subset of CF cache structures are not available, data sets bound to those structures might not be accessible from this system.

Enabling a Data Set for VSAM RLS Processing

For a data set to be opened for VSAM RLS processing, VSAM RLS processing must be available, and the LOG parameter must be specified on the DEFINE CLUSTER or the ALTER CLUSTER command for the data set. Additionally, if the data set is currently assigned to a CF cache structure, that cache structure must be available. If the data set is not currently assigned to a CF cache structure, a cache set must be specified on the storage class, and at least one of the associated CF cache structures must be available.

Monitoring the Coupling Facility for VSAM RLS

This section describes:

- Displaying information about the CF
- Altering the size of CF cache structures
- Altering the size of the CF lock structure
- Changing the IGDSMSxx parameters that support the CF
- Altering the Status of CF Cache Structures and Volumes
- Selecting Data Sets for CF Statistical Monitoring

Displaying CF Information

You can use MVS operator commands as well as ISMF to request CF information.

- Use the MVS DISPLAY XCF command to display information about CFs, connections to the CF, and CFRM policies.
- Use the DISPLAY SMS command or ISMF applications to request information about DFSMS CF structures and SHCDSs.

The following sections describe using the DISPLAY SMS command and ISMF applications in greater detail.

Using the DISPLAY SMS Command

The syntax for the DISPLAY SMS command is shown below.

```
DISPLAY  SMS,SMSVSAM[,ALL]
                      ,CFCACHE(CF_cache_structure_name|*)
                      ,CFLS
                      ,CFVOL(volume_serial_number)
                      ,MONDS(specification_mask|*)
                      ,QUIESCE
                      ,SEP
                      ,SHCDS
```

The following list describes the various forms of the DISPLAY SMS command and what information is displayed:

- `DISPLAY SMS,SMSVSAM[,ALL]`

  displays both the status of the SMSVSAM server and the lock structure mode. If the lock structure is duplexed, composite state information will also be displayed.

  Specify ALL to see the status of all SMSVSAM servers.
- `DISPLAY SMS,CFCACHE(CF_cache_structure_name|*)`

  displays information about CF cache structures. Specify CFCACHE(*) to request information for all CF cache structures. Specify a specific cache structure name to display information about only that cache structure.

- `DISPLAY SMS,CFLS`

  displays information about the CF lock structure. This information includes the lock rate, lock contention rate, false contention rate, average number of requests waiting for locks, the lock structure size, and primary structure information. If the lock structure is in duplex mode, secondary structure information will also be displayed.
- `DISPLAY SMS,CFVOL(volser)`

  displays a list of CF cache structures containing data for the volume specified. Also displays the CF_VOLUME status.
- `DISPLAY SMS,MONDS(specification_mask|*)`

  Specify MONDS(*) to view all the data set specifications eligible for CF statistics monitoring. Use a specification mask to view only a subset of those specifications. You can specify a full or partial data set name, and you must specify at least one high level qualifier. A wild card in the data set name cannot be followed by additional qualifiers.
- `DISPLAY SMS,SMSVSAM, QUIESCE`

  displays active QUIESCEs on the system on which it is issued.
- `DISPLAY SMS,SEP`

  displays the name of the data set separation profile, if one is specified.
- `DISPLAY SMS,SHCDS`

  displays information about SHCDSs. This information includes SHCDS names, sizes and the amount of free space for all the active SHCDSs and their status. It also includes the names of all the spare SHCDSs.

Using ISMF

You can use the ISMF Data Set, Volume and Storage Class applications to display CF information. You can also use ISMF to view the cache sets defined for a specific CDS and the cache structure names associated with each cache set.

Data Set Application: Use the Data Set Selection Entry panel to specify certain attributes and request a list of corresponding data sets and their characteristics. The data sets must be open for VSAM RLS access in order to appear on the list. If you specify CF attributes, ISMF displays the following CF information for each data set matching your criteria:

- CF status indicator

  Indicates whether the data set is in use by RLS processing, whether a forward recovery of the data set is in progress, and whether the sphere has been quiesced for RLS processing.
- CF monitor status

  Indicates whether CF cache structure statistical monitoring is on or off.
- CF cache structure name

  Specifies the name of the CF cache structure in which the data set is stored.
- CF cache set name

  Specifies the name of the CF cache set with which the data set is associated.

You can also use these values as attributes on the Data Set Filter, View, and Sort panels to customize the data set list created by ISMF.

Volume Application: Use the Volume Selection Entry panel to specify certain attributes and request a list of corresponding volumes and their characteristics. You can request that ISMF display CF volume status information, indicating one of the following conditions:

- The volume is enabled for VSAM RLS processing and can be associated with a CF cache structure
- VSAM RLS processing is finishing and no new data can be placed in CF cache structures
- No VSAM RLS data for the volume exists in any of the CF cache structures and no CF cache structures can be assigned to the volume

You can also use the CF volume status attribute on the volume filter, view, and sort panels to further customize the volume list created by ISMF.

Storage Class Application: Use the Storage Class Application Selection panel to request that ISMF display storage class names and CF cache set names associated with a specific CF cache structure name. You can also request to see the CF direct or sequential weights assigned to a specific storage class definition.

In addition, you can use the CF cache set name, CF direct weight, and CF sequential weight attribute on the storage class filter, view, and sort panels to further customize the storage class list created by ISMF.

Control Data Set Application: Use the CDS Class Application Selection panel to request that ISMF display CF cache structure names for all CF cache sets defined for a specific SCDS.

Changing the Size of Coupling Facility Cache Structures

DFSMS uses the initial structure size specified in the CFRM policy each time it connects to a CF cache structure. In each case, the ALTER(YES) keyword is specified, indicating to MVS that this structure can be dynamically reconfigured. To alter the size of a CF cache structure, issue the SETXCF START command, using the following format:

```
SETXCF START,ALTER,STRNAME=CF_cache_structure_name,SIZE=newsize
```

where:

CF_cache_structure_name
: is the name of the cache structure being altered.

newsize
: is the new structure size in megabytes.

  This new size can be larger or smaller than the size of the current CF cache structure, but it cannot be larger than the maximum size specified in the CFRM policy.

MVS automatically starts the alter process in place, without disruptions to the application using the CF cache structure. The alter function does not cross CF boundaries and does not take the place of the rebuild function. If you require a larger structure size than that specified in the CFRM policy, you must activate a new CFRM policy and rebuild the structure. The alter function changes the cache structure in place; you can use the rebuild function to move a cache structure to another CF. See “Defining CF Cache Structures” on page 241 for information on estimating the size of CF cache structures.

Changing the Size of Coupling Facility Lock Structures

You can use the rebuild function to change the size of a lock structure by rebuilding it using a new size. If the new size is too small, then the rebuild is stopped and message IGW322I is displayed. You can then start the rebuild process again using the size recommendations displayed in message IGW322I.

You can also issue the SETXCF START command to change the size of the lock structure. Each time it connects to a CF lock structure, DFSMS uses the initial structure size specified in the CFRM policy. In each case, either the ALTER(YES) or AUTOALTER(YES) keyword is specified, indicating to MVS that this structure can be dynamically reconfigured.

To alter the size of a CF lock structure, issue the SETXCF START command, using the following format:

```
SETXCF START,ALTER,STRNAME=CF_lock_structure_name,SIZE=newsize
```

where:

CF_lock_structure_name
: is the name of the lock structure being altered.

newsize
: is the new structure size in megabytes.

  This new size can be larger or smaller than the size of the current CF lock structure, but it cannot be larger than the maximum size specified in the CFRM policy, and less than its minimum size set by the coupling facility.

MVS automatically starts the alter process in place, without disruptions to the application using the CF lock structure.

If VSAM RLS determines that the lock table space or record table space that results from the alter function will be too small for future locking activity, VSAM RLS issues message IGW322I and continues with the alter function. In this case, you must issue a new alter using the size recommendations displayed in message IGW322I.

The alter function does not cross CF boundaries and does not take the place of the rebuild function. If you require a larger structure size than that specified in the CFRM policy, you must activate a new CFRM policy and rebuild the structure. The alter function changes the lock structure in place; you can use the rebuild function to move a lock structure to another CF.

Changing IGDSMSxx Parameters to Support the Coupling Facility

The SYS1.PARMLIB IGDSMSxx member includes several parameters that support the coupling facility. With the exception of RLSINIT, these parameters apply across all systems in the Parallel Sysplex. The parameter values specified for the first system that was activated in the sysplex are used by all other systems in the sysplex.

The following SYS1.PARMLIB IGDSMSxx member parameters support VSAM RLS:

CF_TIME
DEADLOCK_DETECTION
RLSINIT
RLS_MaxCfFeatureLevel
RLS_MAX_POOL_SIZE
SMF_TIME

You can change the values for these parameters at any time during VSAM RLS processing. The new values are then used by all systems in the Parallel Sysplex, except for RLSINIT parameter values which are used only by the system accessing the changed parmlib member when SMSVSAM is next started.

To change the values of these parameters, use one of the following methods:

- Issue the SETSMS operator command, specifying the parameter with different values.
- Issue the T SMS=*xx* command, where *xx* identifies an IGDSMSxx member where the parameter values are different than those currently in use.

The following list describes the parameters and their values:

{CF_TIME(*nnn*|3600)}
indicates the number of seconds between recording SMF type 42 records with subtypes 15, 16, 17, 18, and 19 for the CF (both cache and lock structures). You can specify a value from 1 to 86399 (23 hours, 59 minutes, 59 seconds). The default is 3600 (one hour).

This keyword sets the interval time for the following SMF 42 subtypes:

SUBTYPE 15 CF storage class average response time

SUBTYPE 16 CF data set average response time

SUBTYPE 17 CF lock structure activity

SUBTYPE 18 CF cache partition summary

SUBTYPE 19 SMSVSAM least recently used statistics summary

{DEADLOCK_DETECTION(*iiii*|15,*kkkk*|4)}
specifies the deadlock detection intervals used by the Storage Management Locking Services.

iiii specifies the length in seconds of the local deadlock detection interval, as a one to four digit numeric value in the range 1-9999. The default is 15 seconds.

kkkk specifies the number of local deadlock cycles that must expire before global deadlock detection is run, as a one to four digit numeric value in the range 1-9999. The default is 4 cycles.

[RLSINIT({NO|YES})]
specify YES if you want the SMSVSAM address space started as part of system initialization or the V SMS,SMSVSAM,ACTIVE command. This value applies only to the system accessed by the parmlib member and is acted upon when SMSVSAM is next started. The default is NO.

[RLS_MaxCfFeatureLevel({A|Z})]
specifies the method that VSAM RLS uses to determine the size of the data that is placed in the CF cache structure.

If you specify **A**, caching proceeds using the RLSCFCACHE keyword characteristics that are specified in the SMS data class that is defined for the VSAM sphere.

If you do not specify a value, or if you specify **Z**, then only VSAM RLS data that have a Control Interval (CI) value of 4K or less are placed in the CF cache structure. The default is **Z**.

Restrictions:

- If **A** is specified for the RLS_MaxCfFeatureLevel parameter, systems lower than V1R3 will not be able to connect to the CF cache structure.
- If a lower-level system is the first system activated in the sysplex, RLS_MaxCfFeatureLevel defaults to **Z**, and all systems will be able to connect to the CF cache structure.
- If the SETSMS command is used to change the RLS_MaxCfFeatureLevel value to **A** on a mixed-level system, the command is rejected and message IGW500I is issued.

[RLS_MAX_POOL_SIZE({nnnn|100})]
specifies the maximum size in megabytes of the SMSVSAM local buffer pool. SMSVSAM attempts to not exceed the buffer pool size you specify, although more storage might be temporarily used. Because SMSVSAM manages buffer pool space dynamically, this value does not set a static size for the buffer pool.

Use SMF 42, subtype 19 records to help you determine the maximum size of the SMSVSAM local buffer pool.

You can specify a two to four-digit numeric value, with 10 as the minimum value. If you specify a value less than 10, the field is set to 10. If you specify a value greater than 1500, SMSVSAM assumes there is no maximum limit. We recommend that you limit the size of the local buffer pool.

The default is **100 MB**.

[SMF_TIME({YES|NO})]
specifies that the following SMF type 42 records are created at the SMF interval time, and that all of the indicated records are synchronized with SMF and RMF data intervals.

SUBTYPE 1	Buffer management statistics
SUBTYPE 2	Cache control unit statistics (IBM 3990-3, 3990-6, IBM RVA, and IBM ESS)
SUBTYPE 15	Coupling facility storage class average response time
SUBTYPE 16	Coupling facility data set average response time
SUBTYPE 17	Coupling facility lock structure activity
SUBTYPE 18	Coupling facility cache partition summary
SUBTYPE 19	SMSVSAM least recently used statistics summary

This allows the customer to merge these SMF records for a specified time period and obtain both the 'system' view and the 'user' view of activity in the interval.

YES is the default. DFSMS creates the specified SMF record when the interval period expires and SMF sends the event notification signal.

If you specify YES, SMF_TIME overrides the following IGDSMSxx parameters: BMFTIME, CACHETIME, CF_TIME.

Changing the State of Coupling Facility Cache Structures and Volumes

You can use the VARY SMS command to control processing for volumes, data sets, or systems.

To alter the state of the specified CF cache structure, issue the VARY SMS command, using the following format:

```
VARY SMS,CFCACHE(CF_cache_structure_name),ENABLE|QUIESCE
```

When a CF cache structure is enabled, VSAM RLS data can be stored in the cache structure. This is the normal state of operations, and is the state the CF cache structure is in after the Parallel Sysplex has been IPLed. When a CF cache structure is quiesced, no VSAM RLS data can be stored in it.

To alter the state of the specified volume as it relates to all CF cache structures, issue the VARY SMS command using the following format:

```
VARY SMS,CFVOL(volser),ENABLE|QUIESCE
```

When a volume is CF-enabled, data contained on this volume can be stored in a CF cache structure. in the cache structure. This is the normal state of operations. When a volume is CF-quiesced, no data contained on it can be stored in a CF cache structure.

Use the VARY SMS CFVOL command if it is necessary to modify a volume without using the VSAM PUT/ERASE macros or DFSMSdss. This ensures that when the modified volume is again made available for VSAM RLS processing, CF cache structures do not contain downlevel data.

Rule: Setting a volume to the CF-quiesced state does not stop SMS from selecting this volume during data set allocation. To stop SMS from selecting this volume, issue the VARY SMS command, using the following format: `VARY SMS,VOLUME(volser),DISABLE`

Selecting Data Sets for Coupling Facility Statistical Monitoring

You can use the VARY SMS command to specify which data set(s) are eligible for CF statistical monitoring. If statistical monitoring is on, SMF TYPE 42, subtype 16 records are produced. Use the following command format:

```
VARY SMS,MONDS(dsname{dsname,...}),{ON|OFF}
```

Select OFF to indicate that the specified data sets are no longer eligible for statistical monitoring. You can specify a full or partial data set name, with at least one high-level qualifier. An asterisk cannot be followed by other qualifiers. You can specify up to 16 data set specifications with each command. This command affects activity for the specified data sets across all systems in the Parallel Sysplex.

Recovering VSAM RLS Processing

This section describes:

- Recovering the CF lock structure
- Recovering a CF cache structure
- Recovering the SMSVSAM address space
- Recovering an SHCDS

Recovering the CF Lock Structure

At VSAM open time, the SMSVSAM address space checks to ensure that the CF lock structure is available. Record-level sharing cannot occur if the CF lock structure is unavailable or has failed.

User-Managed Lock Structure Rebuild

A CF lock structure might fail and need to be rebuilt if the CF lock structure named IGWLOCK00 does not exist, or if it is not connected to the system attempting to open a VSAM data set for record-level sharing. In either case, DFSMS internally

initiates a rebuild queueing all applications using VSAM RLS during the rebuild process. The rebuild is transparent to these applications.

If both the CF lock structure and the system fail, all recoverable data sets which were open for VSAM RLS processing at the time of failure are converted to “lost locks”. The data sets become unavailable to any processing besides recovery processing (such as backouts), and no new sharing is allowed until the recovery processing is complete. A new CF lock structure must be available to perform the recovery processing.

In this case, you can redefine or replace the CF lock structure, correct the problem causing its unavailability, or move the work requiring VSAM RLS to another system which has connectivity to an available CF lock structure.

System-Managed Lock Structure Duplexing Rebuild

Using the system-managed duplexing rebuild function, you can create and maintain a duplex (secondary) copy of the lock structure in case of failure. When the structure is in duplex mode and a failure occurs, VSAM RLS switches from the failed lock structure to the active lock structure, and record-level sharing can proceed. The lock stucture reverts to simplex mode, and continues to operate as it would in user-managed mode (see “User-Managed Lock Structure Rebuild” on page 259.)

When the lock structure is in simplex mode and DUPLEX(ENABLED) is specified for the lock structure in the CFRM active policy, the system attempts to start a system-managed duplexing rebuild if the environment allows. When DUPLEX(ALLOW) is specified, you must use SETXCF START,REBUILD,DUPLEX to manually initiate a system-managed duplexing rebuild.

You can stop system-managed duplexing manually using SETXCF STOP,REBUILD,DUPLEX or by specifiying DUPLEX(DISABLED) in the CFRM active policy.

Restriction: When the lock structure is in duplex mode, user command rebuild requests are rejected. You must use the alter function to change the lock structure.

Related Reading: For more detailed information about the system-managed duplexing rebuild function, see *z/OS MVS Programming: Sysplex Services Guide*.

Recovering a CF Cache Structure

In the event a CF cache structure fails, DFSMS attempts to rebuild it so that it remains available to the data with which it is associated. A CF cache structure is also rebuilt if DFSMS detects a loss of connectivity or an undersized cache structure.

If the rebuilding process is successful and connectivity resumes as before, all opens that were tied to the CF cache structure that failed are automatically reestablished. If the rebuild process fails, but another CF cache structure is defined in the cache set associated with the storage class and connectivity exists to that CF cache structure, then all opens are automatically reassigned to this alternate CF cache structure.

If the rebuild process fails and no alternate CF cache structure is available, any opens currently using the CF cache structure that failed are marked as broken. DFSMS fails the rebuilding process if the new CF cache structure is smaller than the failed structure, or if it does not have the same connectivity. In those cases

where DFSMS cannot rebuild a CF cache structure, the next attempt to open a data set associated with the failed cache structure fails. You might need to redefine the cache structure or correct the connectivity problems.

Recovering the SMSVSAM Server Address Space

If the SMSVSAM server fails, it is automatically restarted. This is a complete reinitialization of the address space and the data space; all connections to prior instances of the SMSVSAM server are invalidated.

Requirement: For SMSVSAM to start, the RLSINIT parameter of the IGDSMSxx parmlib member must be set to YES.

The SMSVSAM server can be automatically restarted up to six times. If the limit is reached, the system issues message IGW418D. You respond to this message by indicating whether the automatic restart facility is to be re-enabled by the system or whether you want to manually enable it. If the automatic restart mechanism has been disabled, use the `VARY SMS,SMSVSAM,ACTIVE` command to restart SMSVSAM and re-enable the automatic restart facility.

Recovering a Shared Control Data Set

You should always run with at least two active and one spare SHCDSs. If a permanent I/O error occurs for an active SHCDS, or if an SHCDS becomes inaccessible from one or more systems, it is automatically replaced by one of the spare SHCDSs. When a system is forced to run with only one SHCDS, it issues a message requesting that you add another active SHCDS and at least one spare SHCDS. If any system does not have access to an SHCDS, all opens for VSAM RLS processing are prevented on that system until an SHCDS becomes available.

The information in SHCDSs is continuously updated. Therefore, backup/restore procedures for SHCDSs are ineffective.

Use the following command formats for the VARY SMS command to add and delete SHCDSs:

To add a new, active SHCDS:	`VARY SMS,SHCDS(`*`SHCDS_name`*`),NEW`
To add a new, spare SHCDS:	`VARY SMS,SHCDS(`*`SHCDS_name`*`),NEWSPARE`
To delete either an active or a spare SHCDS:	`VARY SMS,SHCDS(`*`SHCDS_name`*`),DELETE`

Falling Back from VSAM RLS Processing

When you fall back from VSAM RLS processing, the following occurs: the SMSVSAM address space shuts down permanently on every system in the Parallel Sysplex; the SMSVSAM automatic restart capability is disabled; the lock structure IGWLOCK00 is de-allocated, and all knowledge of SHCDSs and pending subsystem recovery is deleted.

The following sections describe some rules and considerations to follow when falling back from VSAM RLS processing. They also outline the fallback procedure to follow, providing you are only falling back from RLS processing and that you are staying at current product levels. If you are also falling back to different CICS, CICSVR, or DFSMS release levels, see “Understanding the Product Environment for VSAM RLS” on page 236 for a list of rules and considerations to follow.

Fallback Rules and Considerations

Consider and plan for the following situations before you decide to fall back from VSAM RLS processing:

- There might be outstanding recovery for VSAM RLS data sets. The fallback procedure results in the loss of locks protecting back out.
- RLS indicators in the catalog must be reset.
- There might be applications which can only function with VSAM RLS, and which cannot return to a VSAM NSR, LSR or global shared resources (GSR) environment.

Fallback Procedure

Recommendation: Do not use this procedure for normal or abnormal disabling of the SMSVSAM server.

Follow these steps to fall back from VSAM RLS processing:

1. Ensure that there are no outstanding recovery requirements.

   Use the access method services SHCDS LISTRECOVERY or LISTSUBSYSDS command to list all current recovery requirements known to SMSVSAM. Any recovery must be completed prior to continuing with fallback processing, or data integrity is compromised. SMSVSAM is not aware of certain subsystem-related recovery, such as indoubt resolution.

   See *z/OS DFSMS Access Method Services* for a complete description of the access method services SHCDS command.
2. Activate an SMS configuration where all nonblank cache set specifications on storage classes are changed to blank cache set specifications.

   The SMS configuration should not include any cache sets defined in the base configuration. This ensures that the CF is only used for access to data sets which are already bound to a CF cache structure.
3. Ensure that you have no running applications which specify VSAM RLS processing, either specified in an ACB or using JCL.
4. Quiesce all CF cache structures.

   Use the VARY SMS command, as follows:

   ```
   VARY SMS,CFCACHE(CF_cache_structure_name),QUIESCE
   ```

   where:

   CF_cache_structure_name
   is the name of the cache structure being quiesced.

   Issue this command for each CF cache structure in your configuration. Use the D SMS,CFCACHE command to verify that all CF cache structures are quiesced.
5. Reset RLS indicators in all applicable catalogs, using the SHCDS CFRESET command.
6. When you delete RLS information in the catalog, revert to CICS FCT definitions, even if the CICS level remains unchanged

Be sure to complete all the preceding steps before continuing with the procedure to permanently shut down SMSVSAM and delete all knowledge of the lock structure and sharing control.

7. Change the value for the RLSINIT parameter in parmlib member IGDSMSxx to NO in *all* applicable parmlib members and activate the change.

8. Issue the following MVS command to disable the SMSVSAM server:

   ```
   *ROUTE ALL V SMS, SMSVSAM, TERMINATE SERVER
   ```

   Reply `C` to any outstanding IGW418D message. If FORCE SMSVSAM,ARM does not disable the SMSVSAM server, use FORCE SMSVSAM.
9. Issue the following command to complete the VSAM RLS fallback procedure:

   ```
   VARY SMS,SMSVSAM,FALLBACK
   ```

   This command issues message IGW523 to request confirmation. You should first ensure that all SMSVSAM servers are disabled, then respond with:

   ```
   FALLBACKSMSVSAMYES
   ```

   Any other response cancels the command.

   The FALLBACK command forces all lock table connections, and deletes the lock structure and the sharing control group, IGWXSGIS. If any of these steps fail, or if another FALLBACK command is already in process, the command is rejected.

Fallback is complete when message IGW524I is issued to the console that issued the FALLBACK command.

See “Understanding the Product Environment for VSAM RLS” on page 236 for a list of rules and considerations to follow if you are also falling back to different CICS, CICSVR, or DFSMS releases.

Chapter 15. Writing ACS Routines

This chapter provides general-use programming interface and associated guidance information. This chapter is intended to help you to write ACS routines.

ACS routines determine SMS classes and storage groups for all new data set allocations and for data set allocations that occur from converting, copying, recalling, restoring or moving data sets. For objects, ACS routines determine: storage group, when storing them; or storage class and management class, when storing or changing them, or during class transitions. You write ACS routines in the ACS language, which is a high-level programming language. You can write your ACS routines, one for each type of SMS class and one for your storage groups.

After writing the routines, you must translate them into an object form that SMS understands. A successful translation places the ACS object in a specified SCDS. After you activate the configuration contained in that SCDS, ACS routines govern storage management.

When you enter the ISMF Automatic Class Selection Application and select the EDIT option, you are linked to PDF Edit, where you can create or modify ACS routines. Leaving EDIT returns you to the ISMF Automatic Class Selection Application, where you can translate, validate, or test any ACS routine.

This chapter contains four main sections. The first section describes the ACS language constants. The second section describes *read-write variables*. The third section describes *read-only variables*, which the ACS routines use for comparison operations. The fourth section describes the ACS language statements and illustrates the use of the statements in a storage group selection routine.

Constants

You can use four types of constants in ACS routines:

Numeric
A numeric is a string containing up to ten characters, 0 - 9. You can use numerics in comparison operations involving the &NQUAL, &NVOL, and &RETPD read-only variables, which are discussed in “Read-Only Variables” on page 269.

KB, MB
KB and MB are suffixes for numeric constants, such as 200 KB and 10 MB. One KB = 1,024 bytes while one MB = 1,048,576 bytes. Any comparison operation involving the &SIZE and &MAXSIZE read-only variables require that you use KB or MB. They are discussed in “Read-Only Variables” on page 269. The maximum prefix value for KB is 2147483647. The maximum prefix value for MB is 2097151.

Rule: When used for DASD storage, K and M normally mean 1000 and 1000000, not the values used here.

Literal
A literal is a character string, such as 'SYS1.PARMLIB', that is enclosed in single quotation marks. The maximum length of a literal is 255 characters. If you want a literal to contain a single quotation mark, such as PAYROLL'SDATA, then you must specify two single quotation marks: 'PAYROLL''SDATA'.

Mask
A mask is a character string, such as SYS1.*LIB, that is *not* enclosed in single quotation marks. You can use a mask to

© Copyright IBM Corp. 1986, 2002

represent job names, volume serial numbers, or other system values that have a common string of characters, such as all volume serial numbers that begin with IMS. You can also use a mask to represent data set, object or collection names that have a common string of characters.

A mask must begin with an alphabetic character, numeric character (0 - 9), national character ($, @, #), asterisk (*), or percent sign (%). The three characters “*”, “%”, and “.” have special significance in a mask. In addition, the characters ″-″ and ″+″ cannot be used in masks. These characters are reserved for use as continuation characters. The following sections describe the rules for using both the simpler masks and the slightly more involved data set masks. See “FILTLIST Statement” on page 285 for an explanation of the use of the mask characters.

Simple Mask Rules

The following rules apply to the special characters in a simple mask containing a single level name:

- An asterisk, “*”, means that zero or more characters can be present in its place.
- Two or more adjacent asterisks are not allowed within a simple mask.
- A “%” represents exactly one nonblank character. “%%%” represents three character positions.

Simple Mask Examples

TSO*	All names of any length beginning “TSO”
XYZ	All names of any length having three adjacent characters “XYZ”
IMS%%%	All six-character names beginning “IMS”
***%WK%%**	All names where the second and third characters of the last five (or only five) are “WK”

Data Set Mask Rules

The following rules apply to the special characters in a data set mask:

- You can separate data set qualifiers with periods, “.”.
- Each qualifier has a maximum length of eight characters. The maximum length for the entire data set mask is 44 characters.
- A “%” represents a single character position. “%%%” represents three character positions.
- A single “*” by itself indicates that at least one qualifier is needed to occupy that position. A “*” within a qualifier means that zero or more characters can be present.
- A qualifier can be a single “*”
- A “**” means that zero or more qualifiers can be present.
- A “**” cannot appear with any other characters within a qualifier.
- Three or more adjacent “*” are not allowed within a qualifier.
- The read-only variables which cannot be used for comparisons are: &ACSENVIR, &DSNTYPE, &DSORG, &DSTYPE, &LABEL, &RECORG, and &XMODE.

Data Set Mask Examples

SYS1.**	All names where the first (or only) qualifier is “SYS1”
****.OUTLIST**	All names where the last (or only) qualifier is “OUTLIST”
.*.PAYROLL.*.SALARY.	All names with six qualifiers where the third qualifier is “PAYROLL” and the fifth qualifier is “SALARY”
***.%%TEST.*.DATA**	All names with four qualifiers where the second qualifier has six characters ending in “TEST” and the fourth qualifier is “DATA”
****.*ABC*.****	All names where some (or only) qualifier contains the characters “ABC” (or only “ABC”)
LABMGR..DATA**	All names where the first qualifier is “LABMGR” and the last qualifier is “DATA”

Read-Write Variables

You write ACS routines to assign values to read-write variables. You can also use read-write variables as values in comparison operations. These read-write variables *are* case sensitive. The ACS language has four read-write variables:

- &DATACLAS
- &STORCLAS
- &MGMTCLAS
- &STORGRP

The &STORGRP read-write variable should only be used in the storage group routine. It is null on input to the routine unless VOL=REF is specified.

If the read-write variables are explicitly specified by the user, they have an initial value that might be overridden by the ACS routine. If the value is not overridden and the initial name is not defined in a currently active configuration, the allocation fails.

Requirement: You must specify a read-write variable on the PROC statement of the corresponding ACS routine. See “PROC Statement” on page 285 for details.

Each ACS routine can set only its corresponding read-write variable:

- The data class routine can set only &DATACLAS.
- The storage class routine can set only &STORCLAS.
- The management class routine can set only &MGMTCLAS.
- The storage group routine can set only &STORGRP.

The ACS routines assign values to read-write variables using the SET command, which is explained in “SET Statement” on page 286.

In an ACS routine, you can assign an alphanumeric name enclosed in single quotation marks to the read-write variables. Also, you can assign a list of up to fifteen alphanumeric storage group names, each enclosed in single quotation marks, to the &STORGRP read-write variable. If more than one storage group name exists in the list, then each name must be enclosed in single quotes and separated by commas (for example, 'SG1', 'SG2').

Table 22 indicates which read-write variables you can set, which ones you can use for comparisons, and which ones are invalid, for each of the ACS routines:

Table 22. Using Read-Write Variables in ACS Routines

	Read-Write Variable			
ACS Routine	**&STORGRP**	**&MGMTCLAS**	**&STORCLAS**	**&DATACLAS**
Storage group	Set/Compare	Compare	Compare	Compare
Management class	Invalid	Set/Compare	Compare	Compare
Storage class	Invalid	Compare	Set/Compare	Compare
Data class	Invalid	Compare	Compare	Set/Compare

Using Read-Write Variables with Volume Reference (VOL=REF)

When volume reference (VOL=REF) is used, the storage group of the referenced data set is passed to the ACS routines in the &STORGRP read-write variable. However, keep in mind the following:

- The storage group name might not be available if the reference is to a data set on SMS-managed tape. This is because private tapes can be entered into a tape library with a blank storage group name. In this case, the AS routine should use the &LIBNAME read-only variable to determine the storage group for the referenced data set. In this case, the referenced and referencing data sets must reside in the same storage group.
- If the reference is to a new data set, there can be multiple candidate storage groups for the referenced data set and the actual storage group might not have been selected yet. In this case, only the first candidate storage group is passed as input to the ACS routines, and this might not be the storage group in which the referenced data set is eventually allocated.

Using Read-Write Variables with Data Set Stacking

When a data set stacking inconsistency is detected, the ACS routines are re-invoked. When they are available, the following values are passed as input to the ACS routines:

- The storage group of the primary data set
- The storage class of the primary data set
- The management class selected by the previous invocation of the management class ACS routine for the stacked data set
- The data class selected by the previous invocation of the data class ACS routine for the stacked data set

Values might not be available for the following reasons:

- No storage class and storage group is available from the primary data set if it is directed to non-SMS-managed media.
- No management class is available for the stacked data set if it was initially assigned no storage class.
- If SMS was invoked by JES3, it is unable to access the work areas to obtain the storage class, management class, or data class.

- The storage group name might not be available if the primary data set is on SMS-managed tape. This is because private tapes can be entered into a tape library with a blank storage group name. In this case, the ACS routine should use the &LIBNAME read-only variable to determine the storage group for the stacked data set. In this case, the primary data set and the stacked data set must reside in the same storage group.
- If the primary data set is new, there might be multiple candidate storage groups for it and the actual storage group might not have been selected yet. In this case, only the first candidate storage group is passed as input to the ACS routines, and this might not be the storage group in which the primary data set is eventually allocated.

Read-Only Variables

Most ACS variables are read-only. Read-only variables contain data set and system information, and they reflect what is known at the time of the allocation request. You can use read-only variables in comparison operations, but you cannot change their values.

Important: In the data class ACS routine, the &DSNTYPE, &DSORG, &MAXSIZE, &NVOL, &RECORG, and &SIZE variables all default to null if no corresponding value is specified in the JCL.

All of the read-only variables appear in Table 23. The read-only variables *are* case sensitive. The following pages explain the uses of the read-only variables.

Table 23. Read-Only Variables

&ACCT_JOB	&DEF_MGMTCLAS	&GROUP	&MEMN	&PGM
&ACCT_STEP	&DEF_STORCLAS	&HLQ	&MEMNQUAL	&RECORG
&ACSENVIR	&DSN	&JOB	&MSPDEST	&RETPD
&ALLVOL	&DSNTYPE	&LABEL	&MSPARM	&SIZE
&ANYVOL	&DSORG	&LIBNAME	&MSPOLICY	&SYSNAME
&APPLIC	&DSOWNER	&LLQ	&MSPOOL	&SYSPLEX
&BLKSIZE	&DSTYPE	&MAXSIZE	&MSVGP	&UNIT
&DD	&EXPDT	&MEMHLQ	&NQUAL	&USER
&DEF_DATACLAS	&FILENUM	&MEMLLQ	&NVOL	&XMODE

Name **Description**

&ACCT_JOB The accounting information from the JOB statement. (For a description of the indexing function for accounting information, see “Special Functions” on page 281.)

Type: Literal

Max value: 142 characters

&ACCT_STEP

The accounting information from the EXEC statement. This information is refreshed for each step in the job. (For a description of the indexing function for accounting information, see “Special Functions” on page 281.)

Type: Literal

Max value: 142 characters

&ACSENVIR The environment in which the ACS routine was invoked, one of:

RECALL	for data set recall operations
RECOVER	for data set recover operations
RENAME	for data set alter rename operations
RMMPOOL	for DFSMSrmm requests for a storage group name
RMMVRS	for DFSMSrmm requests for a management class name
CONVERT	for data set convert in place operations
ALLOC	for new data set allocations (this is the default)
STORE	OSREQ object store environment
CHANGE	OSREQ object change environment
CTRANS	OSMC object class transition environment
other	installation exit can set its own value before reinvoking ACS

Type: Literal

Max value: 8 characters

&ALLVOL The volume serial numbers specified for data set allocations when &ACSENVIR is not recall or recover. When the environment is recall or recover, &ALLVOL is either the volume serial number on which the data set resided at the time it was migrated or backed up, or the volume serial number specified as the target volume of the recall or recover.

Exception: &ALLVOL is not available to the storage group ACS routine when the environment is recall or recover and when VOLCOUNT(ANY) is specified. See “Special Functions” on page 281 for usage information.

Type: Literal

Max value: 6 characters

The &ALLVOL ACS read-only variable contains the following values when you use VOL=REF:

- ’REF=SD’ (the volume reference is to an SMS-managed DASD or VIO data set)
- ’REF=ST’ (the volume reference is to an SMS-managed tape data set)
- ’REF=NS’ (the volume reference is to a non-SMS-managed data set)

&ANYVOL
The volume serial numbers explicitly specified for the volumes if &ACSENVIR is not recall or recover. When the environment is recall or recover, &ANYVOL is either the serial number on which the data set resided at the time it was migrated or backed up, or the volume serial number specified as the target volume of the recall or recover.

Important: &ANYVOL is not available to the storage group ACS routine when the environment is recall or recover and when VOLCOUNT(ANY) is specified. See “Special Functions” on page 281 for usage information.

Type: Literal

Max value: 6 characters

The &ANYVOL ACS read-only variable contains the following values when you use VOL=REF:

- ’REF=SD’ (the volume is reference to an SMS-managed DASD data set)
- ’REF=ST’ (the volume is reference to an SMS-managed tape data set)
- ’REF=NS’ (the volume is reference to a Non-SMS-managed data set)

&APPLIC
The name of the application associated with the resource owner of the data set (which is set only if RACF is installed and ACSDEFAULTS is YES in IGDSMSxx).

Type: Literal

Max value: 8 characters

&BLKSIZE
The numeric value for the block size specified on the DD statement, dynamic allocation, or TSO ALLOCATE, ranging from 0 - 2147483647 (KB).

Type: Numeric

Max value: 2147483647

&DD
DDNAME in the DD statement of the data set.

Type: Literal

Max value: 8 characters

&DEF_DATACLAS
The data class name associated with the resource owner of the data set (set only if RACF is installed and ACSDEFAULTS is YES in IGDSMSxx).

Type: Literal

Max value: 8 characters

&DEF_MGMTCLAS
The management class name associated with the resource owner of the data set (set only if RACF is installed and ACSDEFAULTS is YES in IGDSMSxx).

Type: Literal

Max value: 8 characters

&DEF_STORCLAS

The storage class name associated with the resource owner of the data set (set only if RACF is installed and ACSDEFAULTS is YES in IGDSMSxx).

Type: Literal

Max value: 8 characters

&DSN

The name of the data set or collection for which ACS processing is taking place. For VSAM data sets, only the cluster name is passed to the ACS routine; the component name(s) are not.

If the data set has an absolute or relative generation number, it is stripped from &DSN The generation number is the low-level qualifier of the data set name.

See Also: For the data set naming rules, see *z/OS MVS JCL Reference*.

Type: Literal

Max value: 44 characters

&DSNTYPE

The data set name type, one of:

EXC	Extended format data set is preferred. The data set allocation is attempted in nonextended format if the necessary system resources for extended are not available.
EXR	Extended format data set is required. The data set allocation fails if unable to allocate in extended format.
HFS	Hierarchical file system data set
LIBRARY	PDSE in SMS-managed storage; partitioned data set in nonmanaged storage
PDS	Partitioned data set
null	No value specified

Type: Literal

Max value: 7 characters

&DSORG

The data set organization, one of:

PS	Physical sequential
PO	Partitioned
VS	VSAM organization
DA	BDAM organization
null	No value specified

Type: Literal

Max value: 2 characters

&DSOWNER

The name of the user or group that owns the data set (set only if RACF is installed).

Type: Literal

Max value: 8 characters

&DSTYPE The data set type, one of:

GDS	One generation data set of a generation data group, or any data set allocated with a relative generation number (such as A.B.C(+1)) or an absolute generation number (such as A.B.C.G0000V00).
PERM	Standard permanent data sets
TEMP	Temporary data sets
null	None of the above.

Type: Literal

Max value: 8 characters

&EXPDT The expiration date in the form of YYYYDDD where YYYY is a year from 1900 - 2155 and DDD is a day in a year from 1 - 366.

Exception: Expiration dates of 99365 and 99366 are considered ″NEVER-SCRATCH″ dates.

Type: Literal

Max value: 7 characters

&FILENUM The value of the FILENUM ACS read-only variable. This variable corresponds to the data set sequence number on the JCL LABEL parameter. The default is 1. This field is optional.

Type: Numeric

Max value: 4 characters

&GROUP The RACF-defined group to which you are connected, or the group specified in the GROUP keyword on the JCL JOB statement. If the environment is recall or recover, &GROUP is set only if the requester of the recall or recover is not a DFSMShsm authorized user. When DFSMShsm invokes the ACS routines, &GROUP is the group associated with &USER

Type: Literal

Max value: 8 characters

&HLQ The high-level (first) qualifier of the data set or collection name.

Type: Literal

Max value: 8 characters

&JOB The job name, the started task name, or the TSO/E userid from the JOB statement, depending on the execution mode (&XMODE). (See “Determining Distributed FileManager/MVS Data Set Creation Requests” on page 282 for Distributed FileManager/MVS usage information.)

Type: Literal

Max value: 8 characters

&LABEL The value of the LABEL ACS read-only variable. This variable corresponds to the label field of the JCL LABEL parameter. Allowable values are NL, AL, SL, NSL, SUL, AUL, BLP, LTM or blank. The default is IBM Standard Label. This field is optional.

Type: Literal

Max value: 3 characters

&LIBNAME The name for the LIBNAME ACS read-only variable, can contain a 1 to 8 character tape library name. This field is optional.

Type: Literal

Max value: 8 characters

&LLQ The low-level (last) qualifier of the data set or collection name.

Type: Literal

Max value: 8 characters

&MAXSIZE The maximum size (in KB or MB) of a new data set. For non-VSAM data sets, the value is &SIZE plus 15 extents. For VSAM data sets, the value is &SIZE plus 255 extents. See “Using Read-Only Variables” on page 277 for more information about the values of &MAXSIZE and &SIZE for VSAM data sets. Also see “Constraints When Using Read-Only Variables” on page 279.

Type: Suffixed numeric

Max value: 2147483647 for KB, 2097151 for MB

&MEMHLQ The high-level (first) qualifier of the object name.

Type: Literal

Max value: 8 characters

&MEMLLQ The low-level (last) qualifier of the object name.

Type: Literal

Max value: 8 characters

&MEMN The name of an object.

Type: Literal

Max value: 44 characters

&MEMNQUAL The number of qualifiers in the object name.

Type: Numeric

Max value: 22

&MSPDEST The destination, specified in data set name format, for a tape management system-driven tape allocation. This value is specified through the AMS pre-ACS installation exit. The data set name format lets you specify a sequence of destinations to be identified, where each qualifier is a specific destination. For example, a data set vaulted first at location OUTD and then sent to OLTS could have an MSPDEST of ’OUTD.OLTS’. The actual values depend on the support provided by your tape management system.

Type: Alphanumeric

Max value: 44 characters

&MSPARM Additional information related to a tape management system-driven tape allocation. This is a variable length field that can be indexed. The value is specified through an external exit.

Type: Alphanumeric

Max value: 256 bytes, including a 2–byte length field for each value specified

&MSPOLICY The name of a management policy associated with tape data for a tape management system-driven allocation. You can use the DFSMSrmm EDGUX100 installation exit to set MSPOLICY to a VRS management value name. You can also set the value of this variable using the SMS pre-ACS installation exit or allow your tape management system to set it using the pre-ACS installation exit.

Type: Alphanumeric

Max value: 8 characters

&MSPOOL A tape pool name associated with the data set being allocated. In a system-managed tape environment with scratch pool support, you can use this variable to specify a default storage group, where the tape storage group is equivalent to the tape pool specified in the variable. If you use the DFSMSrmm EDGUX100 installation exit, you can set this variable to the pool name or prefix determined by the DFSMSrmm scratch pool processing. This variable can also be set through the pre-ACS installation exit.

Type: Alphanumeric

Max value: 8 characters

&MSVGP The group name for mass storage subsystem (MSS) volumes.

Type: Literal

Max value: 8 characters

Note: MVS no longer supports MSS. This variable has no effect.

&NQUAL The number of qualifiers in the data set or collection name.

Type: Numeric

Max value: 22

&NVOL The maximum of the volume count, UNIT count, and number of explicit VOL=SER specifications.

Type: Numeric

Max value: 2147483647

&PGM The name of the program the system is running. (See “Determining Distributed FileManager/MVS Data Set Creation Requests” on page 282 for Distributed FileManager/MVS usage information.)

Type: Literal

Max value: 8 characters

&RECORG The data set record organization, one of:

KS	VSAM key sequenced (KSDS)
ES	VSAM entry sequenced (ESDS)
RR	VSAM relative record (RRDS)
LS	VSAM linear
null	No value specified

Type: Literal

Max value: 2 characters

&RETPD The retention period of NNNN days where NNNN is from 0 - 9999 (JCL maximum). The &RETPD value is calculated from the specified &EXPDT value. If 99365 or 99366, the two ″NEVER EXPIRED″ dates, is specified after 01 January 2000, the calculated value for &RETPD is set to 0. &EXPDT, however, contains the true value of 1999365 or 1999366.

Type: Numeric

Max value: 2147483647

&SIZE The primary amount of space (in KB or MB) requested for a new data set or the amount of space actually used in an existing data set on a DASD volume (see “Constants” on page 265). See “Using Read-Only Variables” on page 277 for more information about the values of &MAXSIZE and &SIZE for VSAM data sets.

Type: Suffixed numeric

Max value: 2147483647 for KB; 2097151 for MB

&SYSNAME Specifies the system name of the system on which the ACS routine is executing. This field is optional. See “Using Read-Only Variables” on page 277 for usage information.

Type: Literal

Max value: 8 characters

&SYSPLEX Specifies the Parallel Sysplex name of the system on which the ACS routine is executing. This field is optional. See “Using Read-Only Variables” on page 277 for usage information.

Type: Literal

Max value: 8 characters

&UNIT IBM-supplied or installation defined generic name for a device type (for example, 3380, SYSDA). For additional possible settings for the &UNIT variable, see “ACS Routine Environments” on page 156.

Type: Literal

Max value: 8 characters

A slash (/) preceding a four digit number represents a unit address (for example, /3380).

When you allocate a tape data set with DISP=MOD and no unit information is specified in the JCL, this variable is blank and SMS might attempt to manage the tape data set as a DASD-resident data set.

&USER The user ID of the person allocating the data set. When DFSMShsm invokes the ACS routines, &USER is either the requestor of the recall or recover, or the user ID of the DFSMShsm address space. If the environment is recall or recover, the variable is set only if the requestor of the recall or recover is not a DFSMShsm authorized user. (See "Determining Distributed FileManager/MVS Data Set Creation Requests" on page 282 for Distributed FileManager/MVS usage information.)

Type: Literal

Max value: 8 characters

&XMODE The execution mode in which the data set is being allocated, one of:

BATCH	Batch execution mode
TSO	TSO execution mode
TASK	A started address space

Type: Literal

Max value: 8 characters

Using Read-Only Variables

The following sections describe various considerations for using read-only variables:

Initializing Read-Only Variables

SMS derives the values of read-only variables before it invokes the ACS routines. The values are based on what is known at the time of the allocation request (for example, if a unit name has been specified on the allocation request, then the &UNIT variable contains the specification).

Exception: &DSNTYPE is not initialized with the DSNTYPE default specified in the IGDSMSxx member in SYS1.PARMLIB.

Using Default Read-Only Variables

Read-only variables are defaulted from data class (if assigned) before storage class, management class, and storage group ACS routines are invoked.

Ensuring Correct Values for &SIZE and &MAXSIZE

If you allocate a data set using the TSO/E ALLOCATE command and you do not explicitly specify space requirements, then &SIZE and &MAXSIZE do not contain the correct values. Instead they both contain a value of zero. If your ACS routines rely on the values of &SIZE or &MAXSIZE in this situation, the data set might be assigned to the wrong class or group.

For a VSAM data set definition, the &SIZE and &MAXSIZE read-only variables reflect the space value specified in the CLUSTER component. If one is not specified in the CLUSTER component, then the space value specified in the DATA component is used. If a space value also is specified for the INDEX component *and* it is of the same type of space unit; for example, both are in tracks, cylinders, KB or MB, it is added to what was specified for the DATA component. If the INDEX component space unit is not of the same type as specified for the DATA component, it is ignored and not added to &SIZE or &MAXSIZE. For DFSORT™ work data sets, &SIZE and &MAXSIZE are zero. In the RECALL environment, &SIZE and &MAXSIZE are zero for empty partitioned data set and PS data sets because the size of the existing data set is not known at the time that DFSMS runs its ACS routines.

Using OAM Read-Only Variables

&MEMN, &MEMHLQ, &MEMLLQ and &MEMNQUAL are used to name an object in a collection.

Restrictions: These read-only variables are valid only when &ACSENVIR is equal to one of the three valid OAM environments (STORE, CTRANS, CHANGE), otherwise, the passed value is nullified by ACS before invoking the ACS routines.

Using &DSN for a Partitioned Data Set

For a partitioned data set, &DSN consists only of the data set name. A member name, if specified, is not be part of the value of the &DSN or &MEMN variables. There is no class selection capability based on member name.

When the Value of &USER Is Null

Not all read-only variable values are significant during the actual operation of any ACS routine. For example, the value of &USER could be null in the storage group selection routine if no user ID had been specified on the JOB statement or determined from the environment.

The user ID (and group ID) is only available if specified using JCL or if RACF (or any other security product) is active. For example, on a RESTORE, if a user ID is specified, the ACERO passed by DFSMSdss does not contain the user ID, and because RACF is not active, DFSMS does not interrogate the ACEE to fill it in.

Using the &SYSNAME and &SYSPLEX Read-Only Variables

Due to connectivity constraints on old DASD and tape, it might be necessary to know the system and Parallel Sysplex where an ACS routine is being executed in order to direct the allocation to a storage group(s) that is accessible from the current system. To support this function, SMS ACS processing uses the &SYSNAME and &SYSPLEX read-only variables.

Do not use the &SYSPLEX variable in ACS routines for JES3 systems as Parallel Sysplex name support is not supported in a JES3 environment. Also, ACS routines for JES3 systems should not rely on the &SYSNAME variable, as the system on which the ACS routines are run (during converter/interpreter) might have nothing to do with the system on which the job is eventually executed.

Read-Only Variables on a Data Set Rename

On a data set rename using &ACSENVIR=RENAME, not all read-only variables are passed to the ACS routines. The following read-only variables *are* passed to the ACS routines:

- &ACSENVIR
- &APPLIC (set by ACS routines)
- &DEF_DATACLAS (set to null by ACS routines)
- &DEF_MGMTCLAS (set to null by ACS routines)
- &DEF_STORCLAS (set to null by ACS routines)
- &DSOWNER
- &DSN
- &DSORG
- &DSTYPE
- &EXPDT
- &GROUP
- &SYSNAME (set by ACS routines)
- &SYSPLEX (set by ACS routines)
- &USER

The following read/write variables are also passed to the ACS routines:

- &DATACLAS
- &MGMTCLAS
- &STORCLAS

WRITE statements are not passed to the ACS routines after a data set rename.

Constraints When Using Read-Only Variables

The following sections summarize some constraints when using read-only variables.

Read-Only Variables Not Allowed in Storage Group Routine

The following read-only variables are *not* allowed in the storage group selection routine:

&ACCT_JOB
&ACCT_STEP
&DD
&JOB
&MSVGP
&PGM
&XMODE

Read-Only Variables in Different Environments

The following read-only variables are *not* passed to any ACS routine unless the environment is specified as &ACSENVIR=ALLOC:

&ACCT_JOB
&ACCT_STEP
&DD
&GROUP
&JOB
&MSGVP
&PGM
&USER
&XMODE

Exceptions:

- If the environment is a non-IBM supported environment, the following read-only variables *are* passed regardless of what is specified in &ACSENVIR:
  &JOB
  &DD
  &PGM
- If the environment specified in &ACSENVIR is RECALL, RENAME, or RECOVER, the following read-only variables *are* passed:
  &GROUP
  &USER

Read-Only Variables Not Available when LIKE is Used

When you use the LIKE parameter on a JCL DD statement or the ALLOCATE command, the following read-only variable values are not available to the ACS routines:

&DSNTYPE
&DSORG
&MAXSIZE
&RECORG
&SIZE

Thus, a data set allocated like a second data set might go into in a different storage group than the second data set.

OAM Read-Only Variables

For restrictions, see “Using OAM Read-Only Variables” on page 278.

Comparison Operators

Comparison operators allow you to determine the relationship between two values. The following comparison operators are allowed:

This:	Means this:
GT or >	Greater than
LT or <	Less than
NG or ¬>	Not greater than
NL or ¬<	Not less than
EQ or =	Equal
NE or ¬=	Not equal
GE or >=	Greater than or equal
LE or <=	Less than or equal

Alphabetic characters are sorted before digits (A-Z come before 0-9). The following comparison is true for all high-level qualifiers alphanumerically greater than “M”:

```
IF &HLQ > 'M' THEN . . .
```

For FILTLIST or mask comparisons, only EQ and NE are valid. See “FILTLIST Statement” on page 285 for details.

Comparison Rules

The following rules apply to comparisons:

1. For a comparison to be valid, one operand must either be a read-only variable or a read-write variable and the other operand must be a constant (any of the four types), a read-only variable, or a FILTLIST name.
2. Numerics are right justified.
3. Literals are left justified and padded with blanks.
4. Type checking is done to ensure that numeric read-only variables are not being compared to characters (literals) and that character (literal) read-only variables are not being compared to numbers. &NQUAL, &NVOL, &SIZE, &MAXSIZE &MEMNQUAL and &RETPD are the only numeric read-only variables.
5. Limited length checking of read-only variables with their maximum length values is performed to ensure that the maximum lengths are not exceeded. For example, the literal to which &DSN is being compared must be no longer than 44 characters. See “Read-Only Variables” on page 269 for maximum lengths.

Boolean Expressions

You can use the following Boolean operators in any ACS routine:

This:	Means this:
AND or &&;	And
OR or |	Or

Expressions in parentheses are processed first. In the following example, the set of OR expressions are processed first, so that

```
WHEN ((CONDITION 1) OR
      (CONDITION 2) OR
      (CONDITION 3) AND
      (CONDITION 4)) SET ...
```

is processed as follows:

```
WHEN
 (CONDITION 1 AND CONDITION 4) OR
 (CONDITION 2 AND CONDITION 4) OR
 (CONDITION 3 AND CONDITION 4) SET ...
```

If you want the AND expression be processed before OR, the AND expression must be included in a set of parentheses. In the following example, AND is processed first so that

```
WHEN ((CONDITION 1) OR
       (CONDITION 2) OR
       ((CONDITION 3) AND
       (CONDITION 4))) SET...
```

is processed as follows:

```
WHEN ((CONDITION 1) OR
      (CONDITION 2) OR
      (CONDITION 3 AND CONDITION 4)) SET...
```

Special Functions

The ACS language provides the following three indexing functions, which allow you to make class selections based on specific details about each data set.

Table 24. Indexing Functions for Read-Only Variables

Selection Routine	Data Set Qualifier	&ALLVOL &ANYVOL	Accounting Information
Data class	Yes	Yes	Yes
Storage class	Yes	Yes	Yes
Management class	Yes	Yes	Yes
Storage group	Yes	Yes	No

Data Set Qualifier Indexing Function

The data set qualifier indexing function lets you index the &DSN and &MEMN variables (for accessing particular qualifiers):

&DSN(1) first qualifier of the data set name

&DSN(3) third qualifier of the data set name

&DSN(&NQUAL)
last qualifier of the data set name

&MEMN(1) first qualifier of the object name

&MEMN(2) second qualifier of the object name

&MEMN(&MEMNQUAL)
last qualifier of the object name

The only accepted values for indexes are numbers (1 through 22) and the read-only variables, &NQUAL and &MEMNQUAL.

&ALLVOL and &ANYVOL Functions

These functions let you compare the volume serial number(s) explicitly specified on input with a comparison variable (for example, a FILTLIST variable). The use of &ALLVOL in a comparison expression returns a true value if ALL of the input volsers satisfy the desired condition. The use of &ANYVOL returns a true value if ANY of the input volsers satisfies the desired condition. For example, let IMS101, IMS102, and TSO191 be the input volsers to the routine in Figure 119:

```
PROC STORCLAS

    IF &ALLVOL = IMS* THEN
       (code left out)
    SELECT
      WHEN (&ANYVOL = TSO*)
       (code left out)
    END
END
```

Figure 119. Example of Constraints when Using the &ALLVOL and &ANYVOL Read-Only Variables

The IF statement is false because not *all* volumes match. While IMS101 and IMS102 satisfy the IMS* mask, TSO191 does not.

The WHEN statement is true, because *any* (at least one) volume does match. While IMS101 and IMS102 fail to satisfy the TSO mask, TSO191 does satisfy the mask.

Related Reading: For a detailed explanation of the IF statement, see “IF Statement” on page 288.

Accounting Information Indexing Function

Allows you to reference specific fields in the JOB or STEP account information. A field is defined as a unit of data which is separated by comma(s) in the account information. The use of &ACCT_JOB(n) and &ACCT_STEP(n), where n is the field number, indicates that indexing is requested. If either &ACCT_JOB or &ACCT_STEP is specified without an index, then the default action is to access the first field of the accounting information (for example, &ACCT_JOB(1) or &ACCT_STEP(1) result by default) .

```
&ACCT_JOB(1)       first field of the JOB accounting information
&ACCT_JOB          first field of the JOB accounting information (default)
&ACCT_JOB(3)       third field of the JOB accounting information
&ACCT_STEP         first field of the STEP accounting information (default)
&ACCT_STEP(5)      fifth field of the STEP accounting information
```

The only accepted values for indexes are numbers from 1 - 71.

Determining Distributed FileManager/MVS Data Set Creation Requests

The ACS routines are used to determine the SMS classes for data sets created by Distributed FileManager/MVS. You can use the &JOB, &PGM, and &USER read only variables to distinguish Distributed FileManager/MVS data set creation requests. The following examples show how each of these variables can be used.

&JOB

The value of the &JOB variable is the job specified in the Advanced Program to Program Communications (APPC) transaction program established by the

installation systems programmer. Figure 120 shows how each &JOB can be used to determine the SMS storage class for a data set being created using Distributed FileManager/MVS:

```
PROC &STORCLASS
 .
 .
 IF &JOB =  'GDEDFM' AND &STORCLAS = '' THEN
   SET &STORCLAS = 'DFMCLASS'
 ELSE
   .
   .
   .
```

Figure 120. Example Showing &JOB Read-Only Variable

&PGM

The value of the &PGM variable is specified in the transaction program profile and is always GDEISASB. Figure 121 shows how &PGM can be used to determine the SMS storage class for a data set being created using Distributed FileManager/MVS:

```
PROC &STORCLAS
 .
 .
 IF &PGM = 'GDEISASB' AND &STORCLAS = '' THEN
   SET &STORCLAS = 'DFMCLASS'
 ELSE
   .
   .
   .
```

Figure 121. Example Showing &PGM Read-Only Variable

&USER

You can designate a specific user ID under which all Distributed FileManager/MVS transaction programs are to run. You can then check the user ID to determine if the data set is being created using Distributed FileManager/MVS. The user ID is specified in the job statement in the Distributed FileManager/MVS transaction program profile.

Figure 122 on page 284 shows how &USER can be used to determine the SMS storage class for a data set being created using Distributed FileManager/MVS:

```
PROC &STORCLAS
 .
 .
 IF &USER = 'user_ID' AND &STORCLAS = '' THEN
   SET &STORCLAS = 'DFMCLASS'
 ELSE
   .
   .
   .
```

Figure 122. Example Showing &USER Read-Only Variable

Statements

This section describes the function and syntax of the ACS language statements that you can use when writing ACS routines.

The continuation characters “+” and “-” allow you to extend literal constants to the next line. To ignore the leading blanks on the following line, use “+”. If you want to include the leading blanks on the next line as part of a literal, then use a “-”. You cannot continue masks, numbers, KB or MB numerics, or keywords.

The maximum number of nesting levels for any combination of ACS statement types is thirty-two. (For example, a nested IF statement is one that appears within an IF statement.)

Comments begin with a slash-asterisk pair, “/*”, and end with an asterisk-slash pair, “*/”.

- You can place comments anywhere within an ACS routine where a delimiter might appear.
- Comments cannot be nested; each comment ends at the first occurrence of an asterisk-slash pair, “*/”.
- Asterisks within the comment statement are treated as a special character. A maximum of 500 asterisks can be included in a single comment statement (multiple comment lines not ended with “*/”). This could affect the number of lines allowed in comment continuation.

The statement types are defined as follows:

PROC	Start of an ACS routine
FILTLIST	Definition of filter criteria
SET	Assigns a value to a read-write variable
DO	Start of statement group
IF	Provides conditional statement execution
SELECT	Defines a set of conditional execution statements
EXIT	Causes immediate termination of the ACS routine and can be used to force allocation failures
WRITE	Sends a message to the end user
END	End of statement group (DO or SELECT) or ACS routine (PROC).

PROC Statement

PROC is the first statement of each ACS routine. It identifies the ACS routine and which read-write variable the routine sets. You can precede the PROC statement with blank lines or comments, but not with other statements. For TSO CLIST coexistence, you can place a blank and then a number, such as 0 or 1, after the PROC statement. The number does not affect ACS language processing. To identify an ACS routine and the value it is to determine, you must specify a read-write variable at the end of the PROC statement. You must also place an END statement at the end of each ACS routine.

PROC *<n> read-write variable*

- *n* is optional and can contain any numeric value.
- *read-write variable* is a mandatory value that can be DATACLAS, STORCLAS, MGMTCLAS, or STORGRP. You can optionally precede the variable with an ampersand, &.

```
PROC 1 DATACLAS
PROC 0 &STORCLAS
PROC &MGMTCLAS
PROC STORGRP
```

END

FILTLIST Statement

The FILTLIST statement is a definition list that you can use when testing variables in an ACS routine. You define the information that you want to include and exclude in the list using the INCLUDE and EXCLUDE keywords. Then you can compare read-only variables to items in the list using IF-THEN and SELECT-WHEN statements, without having to write elaborate AND and OR combinations.

FILTLIST is a definition statement that simplifies comparison operations. It is not an execution statement, and it does not change the value of any variables.

Because a FILTLIST can contain only literal values, you can only compare it to literal read-only variables. This excludes the numerically valued &NQUAL, &NVOL, &SIZE, &MAXSIZE, &MEMNQUAL, and &RETPD read-only variables from FILTLIST comparisons.

You must define a FILTLIST before you reference it in the body of an ACS routine.

FILTLIST *name* [<INCLUDE(*list*)>] [<EXCLUDE(*list*)>]

- *name* is mandatory and can be up to 31 alphanumeric characters in length. You can also use an underscore, _, but it cannot be the first character. In the FILTLIST, you can optionally precede the name with an ampersand, & When referring to the FILTLIST in the body of the routine, you must always precede the FILTLIST name with an ampersand.
- You must specify INCLUDE, EXCLUDE, or both in the FILTLIST statement. If a list item satisfies both the INCLUDE and EXCLUDE criteria, EXCLUDE takes precedence and prevents the item from being included in the list.
- *list* can contain literals, simple masks, and data set masks. You can specify up to 255 entries in the INCLUDE or EXCLUDE lists.

Figure 123 on page 286 shows an example of coding a FILTLIST statement:

```
PROC STORCLAS

  FILTLIST VLIST2 INCLUDE(DBX*, TSO*) EXCLUDE('DBX191', 'TSO256')

  IF &ALLVOL = &VLIST2 THEN
      (some action)

END
```

Figure 123. Example of a FILTLIST Statement

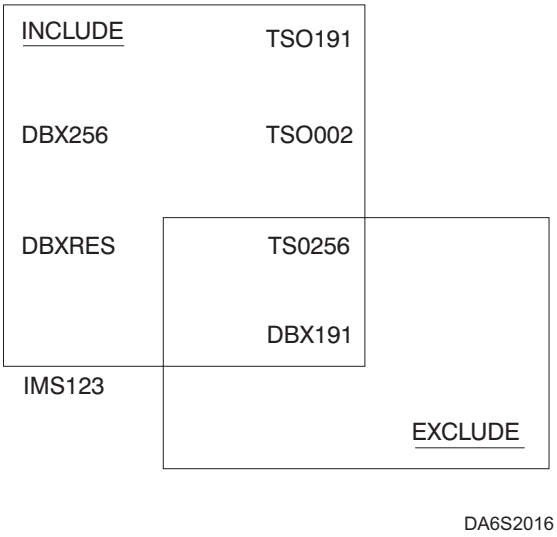


Figure 124. Using the INCLUDE and EXCLUDE

In the environment shown in Figure 124, the value of the IF statement is *true* for any of the following volume serials:

TSO191
TSO002
DBX256
DBXRES

The value of the IF statement is *false* for the following volume serials, because they match the EXCLUDE filter criteria:

TSO256
DBX191

SET Statement

The SET statement assigns values to the read-write variables. The values can be the names of constructs or RACF derived defaults (&DEF_DATACLAS, &DEF_STORCLAS, or &DEF_MGMTCLAS). You can assign one name to &DATACLAS, one name to &STORCLAS, and one name to &MGMTCLAS, but you can assign a list of up to 15 names to &STORGRP

The names can be from one to eight characters long, and they must be enclosed in single quotation marks. The individual names belonging to a &STORGRP must be enclosed in single quotation marks and separated by commas. A name must begin with either an alphabetic or national ($, @, #) character, and the remaining characters can be alphabetic, numeric, or national.

In the body of an ACS routine, you can set only the value of the read-write variable identified in the PROC statement. You cannot set a read-write variable equal to another read-write variable, a FILTLIST variable, or a read-only variable (except for the RACF-derived values). For example, the following is not valid:

```
SET &STORCLAS = &PGM
```

You can assign a null value to any of the read-write variables except for &STORGRP. You can assign a null value by specifying two single quotation marks with nothing between them ('').

```
SET read-write variable = value
```

where:

- *read-write variable* is a mandatory value that can be &DATACLAS, &STORCLAS, &MGMTCLAS, or &STORGRP.
- You can specify EQ in place of the equals sign, =.
- *value* can be one name, a null value, a RACF read-only variable name, or a list of names for the storage group ACS routine. Table 25 summarizes the possible assignments of *value*.

Table 25. Read-Write Variable Assignments

	One Name in Single Quotation Marks	Null Value	RACF Read-Only Variable Name	List of Quoted Names Separated by Commas
Storage group	x			x
Management class	x	x	x	
Storage class	x	x	x	
Data class	x	x	x	

```
SET &STORCLAS EQ 'SCNORM'
SET &STORCLAS = 'SCNORM'
SET &DATACLAS = &DEF_DATACLAS          /* RACF read-only variable */
SET &MGMTCLAS = 'SCRATCH5'
SET &STORGRP = 'SG1','SG2','SG3'      /* list of values          */
SET &DATACLAS = ''                    /* null value assignment   */
```

You must have RACF installed and ACSDEFAULT(YES) specified in the IGDSMSxx to assign the &DEF_DATACLAS, &DEF_STORCLAS, or &DEF_MGMTCLAS. Otherwise a null value is assigned to a read-write variable.

DO Statement

You can group a collection of ACS language statements using a DO statement paired with an END statement. The DO statement can follow an IF-THEN clause, an ELSE clause, or a SELECT-WHEN group.

DO <*group of statements*> **END**

group of statements can consist of zero or more ACS language statements.

Figure 125 shows an example of a DO statement:

```
IF &HLQ='PAYROLL' THEN
 DO
  WRITE 'No Payroll allowed'
  EXIT CODE(1)
 END
```

Figure 125. Example of a DO Statement

IF Statement

Use the IF statement for conditional statement execution. You must always follow an IF statement with a THEN clause. The THEN clause can be a single statement or a DO-END group of statements. You can optionally follow a THEN clause with an ELSE clause. If you specify the ELSE clause, you must follow it with either a single statement or a DO-END group of statements. If you want to specify the ELSE clause but do not want to follow it with an executable statement, follow it with an empty DO-END pair.

If the result of the IF statement comparison is *true*, the THEN clause is executed. If the result is false, the ELSE clause is executed. If you omit the ELSE, processing continues with the next sequential statement after the THEN clause.

IF *relational expression* THEN *clause* <ELSE *clause*>

- *relational expression* can be a single comparison or it can be multiple comparisons joined by Boolean operators.
- THEN is a mandatory keyword.
- *clause* can contain a single statement, a DO-END group of statements, or a SELECT statement.
- ELSE is an optional keyword.
- An ELSE *clause* is mandatory if you specify the ELSE keyword.

Related Reading: For information on what constitutes a valid relational expression, see “Comparison Rules” on page 280.

Figure 126 shows two examples of IF statements:

```
Example of a specified ELSE:          Example of a null ELSE:

IF &DSOWNER = 'BILL' THEN             IF &DSOWNER = 'BILL' THEN
  SET &STORCLAS = 'ONE'                     SET &STORCLAS = 'ONE'
ELSE                                  ELSE
  SET &STORCLAS = 'TWO'                     DO END
                                      SET &STORCLAS = 'TWO'
```

Figure 126. Examples of IF Statements

The statement on the left sets &STORCLAS equal to ONE or TWO, depending on the value of &DSOWNER

The statement on the right sets &STORCLAS equal to TWO, regardless of the value of &DSOWNER If &DSOWNER equals 'BILL', then &STORCLAS is set to

'ONE'. The ELSE clause is skipped, and execution falls to the next statement, which changes &STORCLAS to 'TWO'. If &DSOWNER does not equal 'BILL', execution falls to the ELSE, which results in no assignment to &STORCLAS Then execution proceeds to the next sequential statement, which sets &STORCLAS to 'TWO'.

SELECT Statement

Use the SELECT statement to write conditional statements in sequential form rather than IF-THEN-ELSE form. A SELECT statement consists of a SELECT keyword, one or more WHEN clauses, an optional OTHERWISE clause, and an END statement. You can specify the SELECT statement in one of two forms. In the first form shown below, you include the variable being tested after the SELECT statement. In the second form shown below, you include the variable being tested after the WHEN keyword.

The first true WHEN condition is executed, and the remaining WHEN conditions are ignored. If none of the WHEN conditions is true and there is an OTHERWISE clause, then the OTHERWISE action is taken.

```
SELECT (variable)
  WHEN (value) <action>
                 .
                 .
                 .

 <WHEN (value) <action>>
 <WHEN (value) <action>>
  <OTHERWISE <action>>
END
```

where:

- *variable* can be any read-only or read-write variable.
- At least one WHEN keyword is mandatory.
- *value* can be a constant or a FILTLIST name.
- *action* can be an ACS statement, SELECT group, or a DO END group.
- OTHERWISE is an optional keyword.

```
SELECT (&DSOWNER)
  WHEN ('IBMUSER1') SET &STORCLAS = 'PAYROLL'
  WHEN ('IBMUSER2') SET &STORCLAS = 'TEST'
  WHEN ('IBMUSER3') SET &STORCLAS = 'DEVELOP'
  OTHERWISE      SET &STORCLAS = 'NORMAL'
END
```

SELECT WHEN (*relational expression*) <*action*> . . .

<WHEN (*relational expression*) <*action*>>
<WHEN (*relational expression*) <*action*>>

<OTHERWISE <*action*>> END

- At least one WHEN keyword is mandatory.
- *relational expression* can be a single comparison or it can be multiple comparisons joined by Boolean operators.
- *action* can be an ACS statement, SELECT group, or a DO END group.
- OTHERWISE is an optional keyword.

Figure 127 shows an example of coding a SELECT statement:

```
SELECT
  WHEN (&DSOWNER = 'IBMUSER') SET &STORCLAS = 'PAYROLL'
  WHEN (&DSOWNER = 'IBMUSER2')

     IF &ACCT_JOB = '1234' THEN
       SET &STORCLAS = 'TEST'
     ELSE
       SET &STORCLAS = 'EVERYONE'

  WHEN (&DSTYPE = 'TEMP')  SET &STORCLAS = '
  WHEN (&HLQ    = 'CADAM')   SET &STORCLAS = '
  OTHERWISE                  SET &STORCLAS = 'COMMON'
END
```

Figure 127. Example of a SELECT Statement

EXIT Statement

The EXIT statement immediately terminates the operation of an ACS routine.

EXIT <CODE(*n*)>

- CODE is an optional keyword.
- *n* is an exit code. A nonzero value for *n* causes the subsequent ACS routines to be skipped and the allocation to fail with no explicit value assigned to the read-write variable in the ACS routine. If you do not specify a value for *n*, it assumes the default value of zero.

Figure 128 shows an example of an EXIT statement:

```
PROC STORCLAS
  FILTLIST SECVOL INCLUDE(PAY*, REC*) EXCLUDE('PAYR20', 'REC195')
  FILTLIST VALID_UNITS  INCLUDE('3380','3390','SYSDA','')
  IF &UNIT ¬= &VALID_UNITS THEN

     DO
        SET &STORCLAS = '
        EXIT
     END

  IF &ALLVOL ¬= &SECVOL THEN EXIT CODE(22)

END
```

Figure 128. Example of an EXIT Statement

Note: The example above shows that pre-3380 devices are not supported on DFSMS 1.1 systems. However these earlier devices are still supported on MVS/DFP 3.1.1 and prior systems. The 3330V (MSS) is no longer supported on any system.

If the first IF statement is *true* (&UNIT does not match any unit named in the VALID_UNITS filter criteria), then execution of this ACS routine terminates immediately. Allocation proceeds, because the exit code is zero, the default.

If the second IF statement is *true* (none of the input volumes match the SECVOL FILTLIST criteria), then execution of this ACS routine terminates immediately and the allocation fails. The value for CODE, 22, is set and displayed as part of the allocation failed error message written to the end user.

WRITE Statement

Use the WRITE statement to issue a message to an end user at execution and allocation time. With the WRITE statement, you might notify end users that you are removing a particular storage class, you might inform end users that they lack sufficient authority to use a particular management class, or you might tell an end user that you have moved a tape data set to DASD.

In a TSO/E ALLOCATION environment, the text of the WRITE message displays only if the allocation fails.

Exception: The WRITE message is never displayed for any product or application that uses Dynamic Allocation (for example: DFSMSdss, DFSORT SORTWKnn data sets).

Under certain conditions related to data set stacking, SMS invokes ACS routines more than once. Consequently, you might want to take special care when using WRITE statements in order to avoid duplicates in the job log.

A WRITE message can contain up to 110 characters of text and variables. The message substitutes the value of a variable for a variable name. With the exception of &ANYVOL and &ALLVOL, you can use any of the read-only variables in write statements. All numerical values are in hexadecimal when displayed in write statements.

You must enclose the message in single quotation marks. If you want a single quotation mark to be part of the message, use two single quotation marks to represent it.

You can use continuation characters (+, -) to continue text onto a subsequent line. The closing single quotation mark signifies the end of text.

For example, the following WRITE message:

```
WRITE 'This line''s short.'
```

displays as:

```
This line's short.
```

A nine-character system message id and a single blank character precede your message to the end user. At execution and allocation time, an end user can receive a maximum of five messages. If any more messages are generated, a sixth and final message indicates that additional messages have been generated, but the additional messages are not displayed.

WRITE '*message*'
message is written with the end user's job messages.

Assuming the value of &STORCLAS is 'SC1:', the following WRITE message

```
WRITE 'WARNING - &STORCLAS SPECIFIED (' &STORCLAS ') IS NOT ALLOWED'
```

displays as:

```
IGD01005I WARNING - &STORCLAS SPECIFIED (SC1) IS NOT ALLOWED
```

END Statement

The END statement concludes an ACS routine, a DO group, or a SELECT statement. Figure 129 shows an example of an END statement:

```
PROC STORCLAS
 (source code)
END
```

Figure 129. Example of an END Statement

Sample ACS Routine

Figure 130 illustrates some techniques for using the ACS routines.

```
PROC STORCLAS

/*********************************************************************************/
/* THIS IS THE PRODUCTION SELECTION SPECIFICATION FOR SETTING STORCLAS           */
/*********************************************************************************/

FILTLIST DBVOLS INCLUDE(IMS*,DB2*)                              /* ALL DATABASE VOLUMES */
                EXCLUDE('IMS053','DB2007')

FILTLIST DBJOBS INCLUDE(IMS*,PROD*,ACCT*)                         /* ALL DATA BASE JOBS */

FILTLIST VALID_UNITS
INCLUDE('3330','3340','3350','3375','3380','3390','SYSDA','')      /* VALID UNITS FOR SMS */

IF &UNIT ^= &VALID_UNITS

   THEN DO
           SET &STORCLAS = '
           WRITE 'INVALID UNIT TYPE FOR SMS ALLOCATION'
           EXIT
        END

SELECT

       WHEN (&DSN = SYS1.**)                                          /* SYSTEM DATA */
            SET &STORCLAS = 'SYSTEM'

       WHEN ((&ALLVOL = &DBVOLS) && (&JOB = &DBJOBS))                /* DATABASE DATA */
            SET &STORCLAS = 'DBPOOL'

       WHEN ((&DSN(3) = 'CLEAR') | (&ANYVOL ^= TSO*))                 /* NON-SMS DATA */
            SET &STORCLAS = '

       WHEN (&DEF_STORCLAS ^= '')                                  /* IF DEFAULTS EXIST */
            SET &STORCLAS = &DEF_STORCLAS;

       OTHERWISE SET &STORCLAS = 'COMMON'                             /* ALL OTHER DATA */

END                                                                /* END STORCLAS PROC */
```

Figure 130. Production ACS Routine for Storage Class

The FILTLIST VALID_UNITS INCLUDE statement in Figure 130 on page 292 does not contain the latest devices. Update the FILTLIST VALID_UNITS INCLUDE statement when new devices are installed at your installation.

Restriction: If the null unit that is illustrated in the FILTLIST VALID_UNITS INCLUDE statement in Figure 130 on page 292 is not in the valid DASD units FILTLIST, and a null storage class is assigned to allocations that do not have a valid unit (units in the DASD units FILTLIST), then you cannot manage VSAM data sets allocated using IDCAMS.

Appendix A. Quick Reference to ISMF Commands and Line Operators

The abbreviations in the Application column of these tables represent the following:

ACS Automatic class selection
AG Aggregate groups
CDS Control data set
DC Data class
DS Data set
DV Optical drive
DVOL DASD volume
LA List
LB Optical library
MC Management class
OVOL Mountable optical volume
SC Storage class
SG Storage group
TL Tape library
TVOL Mountable tape volume

Tip: If you specify an equal sign after any DFSMSdss or DFSMShsm line operator, processing occurs in 'last-use mode,' which recalls the last values entered for that particular line operator or command, and DFSMSdfp does not display an entry panel.

Table 26 and Table 27 on page 298 list the ISMF commands and line operators available to storage administrators. See *z/OS DFSMS: Using the Interactive Storage Management Facility* for the line operators and commands available to the end user.

Table 26. ISMF Commands

Command	Minimum Abbreviation	Description	Application	Source
ACTIVATE	AC	Copy the contents of an SCDS into an ACDS and activate it, or activate an existing ACDS.	CDS	DFSMSdfp
ALTER	AL	Alter the use attribute, storage group, shelf location or owner information (or any combination of these) for all tape volumes currently showing in the volume list.	OVOL, TVOL, TL	DFSMSdfp
AUDIT	AU	Verify the location of volumes in 3995 optical libraries or automated tape libraries.	OVOL, TVOL, TL	DFSMSdfp
BOTTOM	BOT	Scroll to the bottom of the entries.	All except ACS and CDS	DFSMSdfp

© Copyright IBM Corp. 1986, 2002

Table 26. ISMF Commands (continued)

Command	Minimum Abbreviation	Description	Application	Source
CANCEL	CA	Return to the previous dialog without performing any of the current dialog functions.	DC, SC, MC, SG, ACS, CDS, AG, LB, DV, LA	DFSMSdfp
CLEAR	CLE	Reset line operator history.	DS, DVOL, DC, SC, MC, SG, AG, LB, DV, OVOL, LA, TVOL, TL	DFSMSdfp
CLEAR ALL	CL ALL	Clear all pages on selection entry panels and filter panels.	DS, DVOL,	DFSMSdfp
CLEAR PAGE	CL PA	Clear the current page on selection entry panels and filter panels.	DS, DVOL, OVOL, LA, TVOL	DFSMSdfp
CLEAR PAGEx	CL PAGEx	Clear a page on selection entry panels and filter panels where x is the page number.	DS, DVOL,	DFSMSdfp
COMPRESS	COM	Reclaim embedded unused space from a list of PDSs.	DS	DFSMSdss
COPY	COP	Copy a list of data sets to a DASD volume of like or unlike device type.	DS	DFSMSdss
DOWN n	DO n	Scroll forward the number of list entries specified by n. DOWN MAX scrolls to the bottom of the entries.	All except ACS and CDS	DFSMSdfp
DSUTIL	DSUTIL	Invoke PDF data set utility functions.	DS	DFSMSdfp
DUMP	DU	Dump data sets to tape, DASD, or mass storage volumes.	DS	DFSMSdss
END	END	Exit the current ISMF function or panel and return to the previous panel.	All	DFSMSdfp
ERTB	ER	Display the ISMF Error Table.	All	DFSMSdfp
FILTER	FIL	Tailor the list to include only specific entries.	DS, DVOL, LA	DFSMSdfp
FILTER CLEAR	FIL C	Clear the filter entries but bypass the entry panel.	DS, DVOL, LA	DFSMSdfp

Table 26. ISMF Commands (continued)

Command	Minimum Abbreviation	Description	Application	Source
FIND	FIN	Find a specific data column.	All except ACS and CDS	DFSMSdfp
FOLD	FO	Extend the data set name data column.	DS	DFSMSdfp
HELP	HELP	Request the help panels associated with the current panel.	All	DFSMSdfp
LEFT	L	Scroll left the specified number of columns.	All except ACS and CDS	DFSMSdfp
LIBRARY	LIBRARY	Invoke PDF library utility.	DS	DFSMSdfp
LISTPRT	LISTP	Prints generated or saved ISMF lists	All that can generate a list	DFSMSdfp
PROFILE	P	Invoke the ISMF profile.	All except the ISMF Profile Application, the ISMF Menu panel or an abend panel	DFSMSdfp
QSAVE	QS	Save query criteria	DS, DVOL	NaviQuest
QRETRIEV	QR	Retrieve query criteria	DS, DVOL	NaviQuest
REFRESH	REF	Display the updated list.	All except ACS and CDS	DFSMSdfp
RELEASE	REL	Free unused space at the end of each of the data sets in a list.	DS	DFSMSdss
RESHOW	RESH	Redisplay all list entries removed by the HIDE line operator.	All except ACS and CDS	DFSMSdfp
RESTORE	REST	Restore data sets that have been dumped by DFSMSdss.	DS	DFSMSdss
RETURN	RETURN	Return to the primary ISMF option menu.	All	DFSMSdfp
RIGHT	RI	Scroll right the specified number of columns.	All except ACS and CDS	DFSMSdfp
SAVE	SA	Save a copy of the current list in the ISPF output table.	DS, DVOL, OVOL, DC, SC, MC, SG, LB, DV, AG, TVOL, TL	DFSMSdfp
SORT	SO	Organize lists based on entries in specific data columns.	All except ACS and CDS	DFSMSdfp
TOP	TOP	Scroll to the top of the entries.	All except ACS and CDS	DFSMSdfp
TSO Commands and CLISTs		Invoke TSO commands and CLISTs.	All	DFSMSdfp

Table 26. ISMF Commands (continued)

Command	Minimum Abbreviation	Description	Application	Source
UP	U	Scroll backward the specified number of entries. UP MAX scrolls to the top of the entries.	All except ACS and CDS	DFSMSdfp
VALIDATE	V	Check the completeness and consistency of an entire SCDS.	CDS	DFSMSdfp
VIEW	VI	Choose the display order of the columns on list panels.	All except ACS and CDS	DFSMSdfp

Table 27. ISMF Line Operators

Line Operator	Minimum Abbreviation	Description	Application	Source
ABACKUP	AB	Specify the parameters for an ABACKUP command and issue the command to DFSMShsm.	AG	DFSMShsm
ALTER	AL	Change the name of a management class, storage class, use attribute, storage group, shelf location, or owner information (or any combination of these) for all volumes associated with a data set or change the use attributes of an SMS class or storage group.	DS, DC, SC, MC, SG, AG, LB, DV, TVOL, OVOL, TL	DFSMSdfp
ANALYZE	ANAL	Examine the device or data on a volume to determine if any errors exist.	DVOL	ICKDSF
AUDIT	AU	Verify the location of optical volumes in IBM 3995 optical and automated tape libraries.	OVOL, LB, TVOL, TL	DFSMSdfp
BROWSE	B	View a sequential data set or a member of a PDS.	DS	DFSMSdfp

Table 27. ISMF Line Operators (continued)

Line Operator	Minimum Abbreviation	Description	Application	Source
BUILDIX	BUI	Change a volume from MVS format VTOC(OSVTOC) to an indexed format VTOC(IXVTOC) or vice versa.	DVOL	ICKDSF
CATLIST	CATLIST	Invoke IDAMS LISTCAT and browse the output.	DS	DFSMSdfp
CLIST	CLI	Call a TSO CLIST.	DS, DVOL, OVOL, TVOL, TL	DFSMSdfp
COMPRESS	COM	Reclaim embedded unused space from a PDS.	DS, DVOL	DFSMSdss
CONDENSE	CON	Free unused space at the end of a data set; compress a PDS.	DS	DFSMShsm
CONTROL	CONT	Reset a device that has been WRITE INHIBITed, reset an indefinite status condition, or clear a fence status of a path or a device or both.	DVOL	ICKDSF
CONVERTV	CONV	Convert DASD volume(s) into SMS or out of SMS.	DVOL	DFSMSdss
COPY	COP	Copy one SMS class or storage group to another SMS class or storage group in the same or in a different SCDS; or copy a data set volume to a DASD volume.	DS, DVOL, DC, SC, MC, SG, AG, LB, DV, TL	DFSMSdfp or DFSMSdss
DEFRAG	DEFR	Reduce free-space fragmentation on a DASD. volume	DVOL	DFSMSdss
DELETE	DEL	Delete an SMS class or storage group, or delete an online, backup, or DFSMShsm-migrated data set.	DS, DC, SC, MC, SG, AG, LB, DV, LA, TL	DFSMSdfp or DFSMShsm
DISPLAY	DI	Display an SMS class and its attributes.	DC, SC, MC, AG, LB, DV, TL	DFSMSdfp
DUMP	DU	Dump a data set or volume to tape or DASD.	DS, DVOL	DFSMSdss

Table 27. ISMF Line Operators (continued)

Line Operator	Minimum Abbreviation	Description	Application	Source
EDIT	E	Edit a sequential data set or member of a PDS.	DS	DFSMSdfp
EJECT	EJ	Eject an optical or tape volume from a library.	OVOL, TVOL, TL	DFSMSdfp
ERASE	ERA	Delete an SMS class or storage group, or delete an online, backup, or DFSMShsm-migrated data set.	DS, DC, SC, MC, SG, AG, LB, DV, LA	DFSMSdfp or DFSMShsm
HALTERDS	HA	Change the number of backup versions of a data set; change frequency of backup.	DS	DFSMShsm
HBACKDS	HBA	Create a backup version of a data set.	DS	DFSMShsm
HBDELETE	HBD	Delete backup versions of a data set.	DS	DFSMShsm
HDELETE	HDE	Delete a migrated data set.	DS	DFSMShsm
HIDE	HI	Remove a list entry from display.	All	DFSMSdfp
HMIGRATE	HM	Migrate a data set to DFSMShsm level one or level two volume.	DS	DFSMShsm
HRECALL	HRECA	Recall a data set that has been migrated by DFSMShsm.	DS	DFSMShsm
HRECOVER	HRECO	Recover a backup version of a data set.	DS	DFSMShsm
INIT	INI	Initialize a volume.	DVOL	ICKDSF
INSPECT	INS	Detect defects in volume track surface.	DVOL	ICKDSF
INSTALL	INST	Install an HDA replacement and physical movement of IBM DASD.	DVOL	ICKDSF
LIST	LI	Retrieve a list that was saved with the SAVE command.	LA	DFSMSdfp
LISTSYS	LISTS	List the systems associated with a given storage group.	SG	DFSMSdfp

Table 27. ISMF Line Operators (continued)

Line Operator	Minimum Abbreviation	Description	Application	Source
LISTVOL	LISTV	List the volumes associated with a given storage group, optical library or tape library.	SG, LB, TL	DFSMSdfp
MESSAGE	MES	Display message text for the last operation performed on a list entry.	All	DFSMSdfp
RAUTH	RAUTH	Provide remote access codes.	DVOL	DFSMSdfp
RECOVER	REC	Recover a backup copy of all the objects on a primary or backup optical cartridge.	OVOL	DFSMSdfp
REFORMAT	REF	Change the volume serial number or owner ID of a volume.	DVOL	ICKDSF
RELEASE	REL	Free unused space at the end of data sets.	DS, DVOL	DFSMSdss
REMAP	REM	Reconstruct inventory in a real IBM 3995 optical library.	LB	DFSMSdfp
REPEAT	=	Repeat the last operator command that was implemented.	All	DFSMSdss DFSMSdfp
RESTORE	REST	Restore data sets that have been dumped by DFSMSdss.	DS, DVOL	DFSMSdss
REVAL	REV	Perform track validation of medial initialization.	DVOL	ICKDSF
SECURITY	SE	Invoke a RACF panel to protect data sets, storage classes, or management classes.	DS, SC, MC, LB, DV, TL	RACF
SETCACHE	SETC	Manage storage control characteristics.	DVOL	DFSMSdfp
SORTREC	SOR	Invoke DFSORT to sort data set records.	DS	DFSORT
STATUS	ST	Allows the display of up to 32 SMS and MVS volume statuses.	DVOL	DFSMSdfp
TSO Commands and CLISTs		Invoke TSO commands and CLISTs.	All	DFSMSdfp
VTOCLIST	VTOCLIST	Invoke IEHLIST LISTVTOC.	DS	DFSMSdfp

Appendix B. SETCACHE Functions and Device Information

This appendix shows you how to:
- Maintain the media characteristics
- Modify DASD storage control
- Provide access to IBM service support representatives at remote locations

To perform these tasks, you enter line operators or list commands on data set or volume lists and then complete data entry panels. Refer to your online help panels or *z/OS DFSMS: Using the Interactive Storage Management Facility*.

Maintaining the Media Characteristics of a Volume

You can use ISMF to examine and manage the characteristics of your data stored on DASD volumes. Eight line operators are available to perform the following media maintenance tasks:

- Inspect a subset of a volume for magnetic surface defects. ISMF also provides options to help you remedy the problem.
- Examine a drive and your data to determine if errors exist. This function helps you distinguish between errors caused by drive problems and errors caused by media problems.
- Initialize a DASD volume for use in an MVS system.
- Change the volume label of a DASD volume. You can specify a new volume serial number or a new owner ID.
- Generate a job stream that is used to change a volume from an MVS format VTOC(OSVTOC) to an indexed format VTOC(IXVTOC) or vice versa.
- Reset a device that has been WRITE INHIBITed, reset an indefinite status condition, or clear a fence status of a path, a device, or both.
- Generate a job stream that is used to perform the procedures necessary for installation, head-disk assembly (HDA) replacement, and physical movement of IBM DASD.
- Perform the track validation functions of medial initialization with the problem determination and data verification functions of the ANALYZE command, and also the INSPECT functions if required.

For information on the ICKDSF functions that ISMF uses to perform these functions see *Device Support Facilities User's Guide and Reference*.

Modifying DASD Storage Control Characteristics

You can use the SETCACHE line operator to control the caching in a storage control unit.

Task	Function	Scope	Action
Manage storage control unit characteristics	SETCACHE	Volume and storage control unit	Modifies the caching status, duplexing status, or causes data to be destaged in the storage control units associated with specific DASD volumes. ISMF performs these functions in the foreground using a TSO command or in the background using the ISMF job submission facility.

© Copyright IBM Corp. 1986, 2002

You can use the SETCACHE line operator from the Volume List panel to modify the storage control unit characteristics. The following characteristics can be modified with the SETCACHE line operator:

- Performance characteristics, which control the caching of read or write requests. You can modify these characteristics to reduce the frequency of access to DASD.
- Availability characteristics, which control the automatic duplication of the device activity onto a secondary volume. You can also modify these characteristics to move the contents of one volume to another while the data is active.
- Resources characteristics, which control how data that is stored in the cache is destaged or discarded. You can directly manage data in cache and nonvolatile storage.

Enter the SETCACHE line operator against a particular volume from the Volume List panel. You can invoke SETCACHE with a PERFORMANCE, AVAILABILITY, or RESOURCES parameter depending on the characteristic you would like to modify. You can also enter SETCACHE without a parameter. On a 3990 Storage Control unit with cache, ISMF takes you to the SETCACHE Features Entry panel. From this panel you can select the characteristic you would like to modify. ISMF takes you to either the PERFORMANCE, AVAILABILITY, or RESOURCES Entry panel. On the Model 3880-13/23, ISMF takes you to the PERFORMANCE Entry panel where you can do your modifications.

The specific features that you can modify depend on the model of the storage control unit that you are working with. You must also have storage administration authorization to use the SETCACHE line operator.

In order to use the SETCACHE function under ISMF, you must modify the IKJTSOxx member in SYS1.PARMLIB. IDCAMS must be added to AUTHPGM NAMES, and SETCACHE must be added to AUTHCMD NAMES.

Modifying Caching Characteristics

Use the Setcache Performance Entry panel to modify the caching characteristics of the storage control units. If you are using a Model 3990 Model 3 Storage Control, you have support for the caching of both read and write requests. If you are using a 3880-13 or 3880-23 Storage Control, you have support for the caching of read requests. In the terminology of the SETCACHE line operator, the caching of read requests is controlled by the READ CACHE or READ and SYSTEM CACHE functions. The caching of write requests is controlled by the DASD FAST WRITE, NON-VOLATILE STORAGE (NVS), and CACHE FAST WRITE functions.

Restrictions: The cache/DFW of the IBM Enterprise Storage Server (ESS) is on by default, and you are not allowed to modify it. In addition, the ESS does not support the dual-copy function. You receive a message, with a condition code of 12, indicating these changes if you issue any of the cache/DFW or dual-copy commands.

Read Caching/Read and System Caching

You can establish or terminate the READ CACHE or READ and SYSTEM CACHE functions to control the caching of read requests. Read caching is performed on a track basis. When you need data from a volume, a read request is made to the storage control unit The storage control unit reads the data from the volume and sends it back to you. At the same time, a copy of all the data on the track that the storage control unit accesses is saved in cache that resides in the storage control

unit. Any subsequent requests that you make for data on the track can be satisfied from the cache. Read performance is improved when the data for your read requests come from the cache.

You can specify that the storage control unit perform this type of read caching on a volume level. Caching can be turned on and off for individual volumes. This feature gives you greater flexibility to manage the volumes that are using the caching resources.

Write Caching

If your storage control unit supports the caching of write requests, you can use the SETCACHE line operator to establish or terminate the DASD FAST WRITE and CACHE FAST WRITE functions. Read and system caching must be active in order for fast write (both DASD and CACHE) to be in effect.

DASD FAST WRITE provides support for the caching of write requests. NON-VOLATILE STORAGE (NVS) provides backup storage for this function. When you write data out to a volume, a write request is made to the storage control unit. The storage control unit accepts the data and puts it in the storage control unit cache. Later, the storage control unit schedules the data for writing.

When the data is stored in the storage control unit cache, it is also written to NVS. NVS is a buffer backed up by battery power that can maintain data up to 48 hours. NVS provides insurance against the loss of data written to the storage control unit cache and not yet sent to DASD.

The CACHE FAST WRITE feature allows part of the subsystem storage to be set aside as work space. Data written out in CACHE FAST WRITE mode is not staged for writing to DASD unless you make an explicit request or the system needs to reuse some of the cache space. CACHE FAST WRITE is for temporary work files only because data in cache is eventually discarded.

Modifying Duplexing Characteristics

Use the Setcache Availability Entry panel to manage the availability of a duplex pair of devices that control the duplication and recovery of data. The AVAILABILITY characteristics affect only data that is on DASD.

To establish a duplex pair, you must specify:
- A primary and a secondary device
- The type of synchronization between the two devices

You can specify the rate of copy in the synchronization. Once duplexing is established, the storage control unit writes data to the primary device and simultaneously places a copy of it in cache along with control information in NVS. At a later time, the data in cache is written out to the secondary device, and the control information in NVS is updated.

If duplexing is interrupted, the dual copy pair is suspended. The storage control still writes information to NVS, but data is written to only the primary device. When duplexing is resumed, the secondary device is placed back in synchronization by the storage subsystem.

You also can use the Setcache Availability Entry panel to move the contents of one device to another. The panel allows you to establish a duplex-pair between two devices to copy data. The pair is immediately broken when all data has been copied. The original device is then available for maintenance.

Modifying Destaging Characteristics

Use the Setcache Resources Entry panel to directly manage the data that has already been written to the cache and NVS. This function of the SETCACHE line operator allows you to:

- Destage data
- Discard pinned data
- Set all the volumes and the subsystem back to their default status

Once data has been written to the cache or NVS, the Resources Entry panel allows you to write the data to DASD. If this destaging process fails, the data is pinned in cache or NVS. The Resources Entry panel allows you to discard this pinned data. All the volumes attached to a storage subsystem along with the subsystem itself can be set back to their default status by reinitializing the storage subsystem.

Restriction: reinitializing a subsystem causes all duplex pairs to be lost and any fast write data in cache and NVS to be discarded.

Submitting Jobs

When you specify the SETCACHE line operator, ISMF takes you through the entry panels you need to complete given the functions you choose to perform. ISMF eventually asks you to specify whether the job is performed in the foreground or the background. In the foreground, ISMF uses the IDCAMS TSO SETCACHE support. ISMF subsequently redisplays updated versions of the entry panels with their new values after the job has executed. The list panels are not updated unless a REFRESH is done. For some functions, execution in foreground can occupy your TSO terminal for 20 minutes or more. In these cases, execution in the background is recommended.

When you specify that the SETCACHE job be performed in background, ISMF displays the IDCAMS Job Submission Entry panel. From this panel, you can submit the job or specify a data set where ISMF saves the job. The Job Submission Entry panel also optionally allows you to change the JCL that ISMF uses to run the job. At each level of the SETCACHE panels, the ISMF online help provides detailed descriptions for each of the fields.

Once ISMF takes you into the Setcache Entry panel, the online help provides detail descriptions for each of the fields.

Providing Remote Authorization Codes

The RAUTH line operator provides an easy way to acquire the remote access authorization codes for your IBM 3990 Storage Control units:

Task	Function	Scope	Action
Provide remote access codes	RAUTH	Volume	Invokes a RAUTH display panel that shows the remote access authorization codes.

When you enter the RAUTH line operator from the Volume List panel, ISMF returns passwords on the RAUTH display panel.

Each password is valid for one hour. Once the password is displayed, ISMF does not display the password again. If you specify RAUTH again, ISMF returns new passwords.

Using these passwords, an off-site IBM Service Representative can log on to all models of the IBM 3990 Storage Control unit family (Models 1, 2, 3, and 6). The Service Representative can then provide help diagnosing and correcting problems.

You must have storage administrator authorization to use the RAUTH line operator. Because the standard support facilities of ISPF are available, for example, printing the screen, it is your responsibility to maintain the security of the passwords. Also, RACF DASDVOL alter authority is required over the volume the request is made against. ISMF online help provides a description of each of the fields on the RAUTH display panel.

Appendix C. Space Utilization and Capacity Planning

You can collect information on active and inactive data set space utilization and capacity planning by using the ISMF Data Collection application to produce a DCOLLECT job. DCOLLECT provides measurement data in a sequential data set (also called a “flat” file), which can be used as input to other applications such as billing and report formatting.

You can use it to produce measurement data on: active data sets, which are data sets that have not been backed up or migrated; inactive data sets, which are data sets that *have* been backed up or migrated; capacity planning, which concerns volume capacity and usage; and volumes. Information can be obtained on the following:

- Active data sets
- Volumes
- Migrated data sets (DFSMShsm)
- Backup data sets (DFSMShsm)
- DASD capacity planning (DFSMShsm)
- Tape capacity planning (DFSMShsm)

DCOLLECT is an access method services function.

Using the Data Collection Application

You can use ISMF panels to generate the job control language for the access method services DCOLLECT command. The job is expected to be executed while the DFSMS environment is active. To use these panels, select the Data Collection Application option on the ISMF Primary Option Menu for Storage Administrators.

Figure 131 on page 310 shows the page 1 of the Data Collection Entry panel. The fields on the entry panel for the Data Collection application are primed to collect data only for active data sets and volumes. You can also collect information on migration data, backup data and capacity planning data. You can specify Y, yes, in any combination of the fields under SELECT DATA COLLECTION OPTIONS.

© Copyright IBM Corp. 1986, 2002

```
  Panel  Utilities  Scroll  Help
-------------------------------------------------------------------------------
DGTDAD01                  DATA COLLECTION ENTRY PANEL             Page 1 of 3
Command ===>

Select Data Collection options:
  Data Set Information  . . . Y           (Y or N; Y requires volume(s) or
  Volume Information  . . . . Y           storage group(s) on next page)
  Migration Data  . . . . . . N           (Y or N)
  Backup Data . . . . . . . . N           (Y or N)
  Capacity Planning Data  . . N           (Y or N)
  SMS Data  . . . . . . . . . N           (Y or N)
Specify Output Data Set:                  (1 to 44 Characters)
  Data Set Name . . . . . . .
  Optional Password . . . . .             (Ignored if SMS-managed data set)
  Replace Contents  . . . . . N           (Y or N)
  Number of Data Sets . . . . 1           (1 to 99999999; new data set only)
Specify Input Data Set:                   (1 to 44 Characters)
  Migration Data Set Name . .
  Backup Data Set Name  . . .
  CDS Name  . . . . . . . . . 'ACTIVE'

Use ENTER to Perform Selection; Use DOWN Command to View next Entry Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 131. Entry Panel for the Data Collection Application, Page 1 of 3

The following list explains the fields on page 1 of the Data Collection Entry panel.

DATA SET INFORMATION
If you want information collected on active data sets selected by storage group or volume, specify Y, yes, in this field. Otherwise, specify N, no. The default is Y.

VOLUME INFORMATION
If you want to collect information on active volumes selected by storage group or volume, specify Y, yes, in this field. Otherwise, specify N, no. The default is Y.

MIGRATION DATA
If you want to collect information on migration data, specify Y in this field. Otherwise specify N, no. If you specify Y, you must specify a migration data set name in the MIGRATION DATA SET NAME field. The default is N.

BACKUP DATA
If you want to collect information on backup data, specify Y in this field. Otherwise specify N, no. If you specify Y, you must specify a backup data set name in the BACKUP DATA SET NAME field. The default is N.

CAPACITY PLANNING DATA
If you want to collect information on capacity planning, specify Y in this field. Otherwise specify N, no. If you specify Y, you must specify both a migration data set name and a backup data set name. The default is N.

SMS DATA
Specify Y, yes, in this field if you want to collect SMS information. The default is N.

DATA SET NAME
You must enter the name of an output data set in this field. DCOLLECT puts the information it collects in the data set you specify. The data set can either be an existing data set or a data set that is allocated during the data collection job while DFSMS is active.

OPTIONAL PASSWORD
If the data set is not system-managed and it has a password, you must specify the password in the OPTIONAL PASSWORD field. Password protection is less secure than RACF protection. The data set must be cataloged. This field is ignored for system-managed data sets.

REPLACE CONTENTS
If you want DCOLLECT to overlay any data which is already contained in the output data set, specify Y, yes, in the REPLACE CONTENTS field. If you want DCOLLECT to append newly collected data to the data already contained in the output data set, specify N, no, in the REPLACE CONTENTS field. N, no, is the default, that is, data already contained in the output data set is not erased when the job is submitted. If the data set has never been written in, it does not matter whether you specify Y or N.

NUMBER OF DATA SETS
If you are specifying an output data set that is to be allocated during the data collection job, you should estimate the total number of data sets which reside on the volumes about which you are collecting data. Specify this number in the NUMBER OF DATA SETS field. This number is used to estimate the size of the output data set. If you save the JCL generated, instead of submitting it directly from this application, you can edit the DD statement for the output data set to change the space calculation.

MIGRATION DATA SET NAME
If you are collecting migration data or information for capacity planning, you must specify a data set name in the MIGRATION DATA SET NAME field. This data set must contain migration information. For DFSMShsm, this data set is the Migration Control Data Set (MCDS) for either the system you are running on or the system about which you want information. This data set must be accessible to the system running the DCOLLECT job.

BACKUP DATA SET NAME
If you are collecting backup data or information for capacity planning, you must specify a data set name in the BACKUP DATA SET NAME field. This data set must contain backup information. For DFSMShsm, this data set is the Backup Control Data Set (BCDS) for either the system you are running on or the system about which you want information. This data set must be accessible to the system running the DCOLLECT job.

When you are collecting capacity planning information, you must specify migration and backup data sets that are on the same DFSMShsm system.

If you are collecting information on active data sets or volumes, use the DOWN command to reach the next page of the Data Collection Entry panel shown in Figure 132 on page 312.

```
  Panel  Utilities  Scroll  Help
--------------------------------------------------------------------------------
                           DATA COLLECTION ENTRY PANEL              Page 2 of 3
Command ===>

Specify Pool Storage Groups (Full Name)
   ==>                                  ==>
   ==>                                  ==>
   ==>                                  ==>
   ==>                                  ==>
   ==>                                  ==>
Specify Volumes (Full, Partial, ******, or *)
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>
Use ENTER to Perform Selection; Use UP Command to View previous Entry Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 132. Entry Panel for the Data Collection Application, Page 2 of 3

On this panel, you must specify one or more volumes or storage groups about which DCOLLECT is to collect information.

If you specified Y, yes, in the DATA SET INFORMATION or VOLUME INFORMATION fields on the first page of the DCOLLECT entry panel, you must specify at least one volume or storage group on this panel.

SPECIFY VOLUMES

In this field, you can specify the names of the volumes from which information is collected. The information collected can be about the volumes themselves or about the active data sets they contain. You can specify a full volume name (for example, SYSTSO), a partial volume name (for example, SYST*), an asterisk to indicate all online volumes, or six asterisks to indicate the system residence (SYSRES) volume.

DCOLLECT ignores repeated entries. If you specify a volume more than once, DCOLLECT collects information on that volume only once. This includes volumes belonging to storage groups that you have specified. If you specify a single asterisk, DCOLLECT ignores all other entries in the SPECIFY VOLUMES field. Although you can specify only 50 volumes on this panel, you can edit the job control language produced to specify up to 255 separate volumes.

SPECIFY POOL STORAGE GROUPS

In this field, you can specify the names of pool storage groups from which information is collected. Information from all of the volumes in a specified storage group is collected. Enter only pool storage group names. Only pool storage groups are appropriate for data set or volume information collection. The information collected can be on the volumes themselves or on the active data sets they contain. You must specify the full name of a storage group. You cannot specify a partial name or an asterisk for a storage group name.

Although you can specify only 10 storage groups on this panel, you can edit the job control language produced to specify up to 255 storage groups.

If you are collecting information only on active data sets, DCOLLECT collects information on the active data sets on the volumes and, in the storage groups you specify. It does not collect information on other data sets on those volumes and in those storage groups.

If you are collecting information only on active volumes, DCOLLECT collects information on the volumes and in the storage groups you specify. It ignores information about the data sets on those volumes.

If you want to specify volumes to exclude, use DOWN command to reach the last page of the Data Collection Entry panel shown in Figure 133.

```
  Panel  Utilities  Scroll  Help
--------------------------------------------------------------------------------
                          DATA COLLECTION ENTRY PANEL              Page 3 of 3
Command ===>

Specify Exclude Volumes (Full or Partial)
   ==>           ==>           ==>           ==>           ==>
   ==>           ==>           ==>           ==>           ==>









Use ENTER to Perform Selection; Use UP Command to View previous Entry Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 133. Entry Panel for the Data Collection Application, Page 3 of 3

When you have finished entering values on the Data Collection Entry panel, press ENTER to display the DCOLLECT Job Submission Entry panel. On this panel, you can choose to submit the job you just specified, or save the job to a data set, which you can then edit. You can edit the job statement on this screen before submitting the job.

The JCL produced uses the access method services execute statement that you have already set up in the profile as a default. You can specify that you want to edit the execute statement on the Data Collection Job Submission Entry panel. When you press ENTER, you can edit the execute statement.

Appendix D. Using Data Set Separation

Data set separation allows you to designate groups of data sets in which all SMS-managed data sets within a group are kept separate, on the physical control unit (PCU) level, from all the other data sets in the same group.

Overview

To utilize data set separation, you must create a data set separation profile and specify the name of the profile to the base configuration. During allocation, SMS will attempt to separate the data sets listed in the profile.

A data set separation profile contains at least one data set separation group. Each data set separation group contains a fail level keyword value and a list of data set names to be separated from each other during allocation. The fail level keyword value specifies *required* or *preferred* data set separation.

Restriction: You cannot use data set separation when allocating non-SMS-managed data sets or during use of full volume copy utilities such as PPRC.

Recommendation: Use data set separation only for a small set of mission-critical data for the following reasons:

- An existing data set might be allocated to one or more PCUs from which the current allocation is to be separated. In this case, SMS either rejects all volumes on the PCUs from the SMS volume preference list or places them at a less-preferred position. Volume rejection might drastically reduce the number of eligible volumes or result in allocation failures if SMS rejects all volumes.
- Data set separation can affect system performance. See the performance concerns listed in “Factors Affecting Code Path Length” on page 319.

Related Reading: For information about specifying the name of the data set separation profile to the base configuration, see “Specifying the DS Separation Profile” on page 31.

The following table lists the topics that are contained in this section:

Topic	Refer to page
Syntax for Creating a Data Set Separation Profile	316
Steps for Creating a Data Set Separation Profile	317
Creating Multiple Data Set Separation Profiles	318
Using Data Set Separation with Generation Data Groups and Striping	318
Environmental Conditions Affecting Data Set Separation	319
Factors Affecting Code Path Length	319

© Copyright IBM Corp. 1986, 2002

Syntax for Creating a Data Set Separation Profile

A data set separation profile contains at least one data set separation group. Use the following syntax to create a data set separation group:

```
SEPARATIONGROUP|SEP
FAILLEVEL|FAIL ({PCU|NONE})
DSNLIST|DSNS|DSN (data set name{,data set name,...})
```

You can specify either *required* or *preferred* data set separation by using one of the following values for the `FAILLEVEL` keyword:

FAILLEVEL(PCU)
A `FAILLEVEL` value that indicates that separation on the PCU level is *required*. SMS fails the allocation if the requested data set cannot be separated on the PCU level from other data sets that are listed in the `SEPARATIONGROUP`. An allocation failure message is issued that indicates the number of volumes that were rejected by the data set separation profile.

FAILLEVEL(NONE)
A `FAILLEVEL` value that indicates that separation on the PCU level is *preferred*. SMS allows the allocation if the requested data set cannot be separated on the PCU level from other data sets that are listed in the `SEPARATIONGROUP`. SMS allocates the data sets on the same PCU and issues an allocation message that indicates that separation was not honored for a successful allocation. During volume selection, SMS treats all volumes on the same PCU level as less-preferred.

Data set names listed in a data set separation group must not contain quotes or wild cards, and must follow the naming convention described in *z/OS MVS JCL Reference*. You can specify the same data set name in multiple data set separation groups.

Writing Comments
Use one of the following two methods for entering comments into a data set separation group:

! SMS ignores all text on a line following the comment character (!). To continue the comment on the next line, you must also precede the next line with the comment character. For example:

```
! This illustrates a comment which continues
! on the next line.
```

/* */ SMS ignores all text within the comment start indicator (/*) and comment end indicator (*/). For example:

```
/*---------------------------
  This illustrates a comment which continues
  on the next line.
  ---------------------------*/
```

Indicating Continuation and Termination
Use the following notational conventions to indicate continuation or termination of a data set separation group:

- Use this character to continue a SEPARATIONGROUP statement onto a following line. You cannot continue a keyword or keyword value in the middle of the keyword or keyword value. You must specify continuation characters outside of comments. For example:

```
SEP FAIL(PCU)-
/*SMS Control Data Sets*/-
```

; You must use this termination character to end a SEPARATIONGROUP statement. For example:

```
SEP FAIL(PCU)-
DSNLIST(SMS.PROD.SCDS-
,SMS.PROD.ACDS);
```

Example: The following example of a data set separation profile contains two comments and two data set separation groups:

```
/*----------------
  SMS CONTROL Data Sets
  --------------*/
SEPARATIONGROUP-
  FAILLEVEL(NONE)-
  DSNLIST (SMS.PROD.SCDS, SMS.PROD.ACDS, SMS.PROD.COMMDS);

/*----------------
JES CHECKPOINT Data Sets
  --------------*/
SEP FAIL(PCU) DSNS(SYS1.JESCKPT1,-  !primary
                 SYS1.JESCKPT2,-    !secondary
                 SYS1.JESCKPT3);    !tertiary
```

Steps for Creating a Data Set Separation Profile

Before you begin: Read “Overview” on page 315. You also need to know the syntax described in “Syntax for Creating a Data Set Separation Profile” on page 316, and the following requirements and recommendations:

Requirements:

- Define the data set separation profile as a PDS member or sequential data set.
- Ensure that the profile is cataloged.
- Use a fixed record format or a fixed-block record format by specifying RECFM=FB.

Recommendation: Use the DISP=OLD disposition to modify the profile. The data in the profile might be corrupted if you use the DISP=SHR disposition.

Perform the following steps to create a data set separation profile:

1. Enter comments that describe the first data set separation group. This step is optional.
2. Enter the keyword `SEParationgroup`.
3. Enter the keyword `FAILlevel`.
4. Enter one of the following FAILLEVEL values to specify which action SMS must perform if data set separation cannot be achieved:
   a. `(PCU)` specifies that data set separation is a *required* condition.
      `FAILlevel (PCU)`
   b. `(NONE)` specifies that data set separation is a *preferred* condition.
      `FAILlevel (NONE)`
5. Enter the keyword `DSNLIST`.
6. Enter the names of data sets that you want SMS to keep separate from each other during allocation.
   `DSNlist(`*`data set name`*`{,`*`data set name`*`,...}`)`

You can also specify the DSNLIST keyword as DSNS.

7. If your data set separation profile contains more than one data set separation group, repeat steps 1 on page 317 through 6 on page 317.

When you are done, you have created a data set separation profile that looks similar to one of the following examples.

Data Set Separation Profile Examples:

- **Example 1:** Data set separation is specified as *required*.

```
/*-------------------------
  PROD CHECKING Data Sets 1
  -------------------------*/
SEPARATIONGROUP-
 FAILLEVEL(PCU)-
 DSNLIST(PROD.CHECKING.ACCT, PROD.CHECKING.HISTORY);

/*------------------------
  PROD SAVINGS Data Sets 1
  ------------------------*/
SEP FAILLEVEL(PCU)-
 DSNLIST(PROD.SAVINGS.ACCT, PROD.SAVINGS.HISTORY);
```

- **Example 2:** Data set separation is specified as *preferred*.

```
/*-------------------------
  PROD CHECKING Data Sets 2
  -------------------------*/
SEPARATIONGROUP-
 FAILLEVEL(NONE)-
 DSNLIST(PROD.CHECKING.ACCT, PROD.CHECKING.HISTORY);

/*------------------------
  PROD SAVINGS Data Sets 2
  ------------------------*/
SEP FAILLEVEL(NONE)-
 DSNLIST(PROD.SAVINGS.ACCT, PROD.SAVINGS.HISTORY);
```

Creating Multiple Data Set Separation Profiles

You can create multiple data set separation profiles, but you can specify only one in an SMS base configuration (SCDS).

If you have created multiple SMS configurations, each configuration can share the same data set separation profile, or each can specify a different data set separation profile. Once an SMS configuration is activated, all systems within the SMS complex will share the same data set separation profile that is specified in the base configuration.

SMS reads the data set separation profile when SMS initializes with an active configuration, when SMS restarts, and when a configuration is activated or switched.

Using Data Set Separation with Generation Data Groups and Striping

Data set striping can use multiple controllers when multistriped. When using data set separation that references multistriped data sets, ensure that the storage group ACS routines select storage groups that contain a sufficient number of PCUs.

Generation data groups (GDGs) that are listed in the data set separation profile must specify the absolute generation and version number (for example, A.B.C.G0001V00). Data set separation does not support relative generation numbers and GDS base names.

When specifying separation of GDGs, ensure that the number of generations that reside on pool volumes does not exceed the number of available PCUs.

Recommendation: Migrate older GDG generations to allow a sufficient number of PCUs for the nonmigrated GDG generations.

Related Reading:

- For more information about GDGs, see “GDG Management Attributes” on page 78.
- For more information about data set striping, see “Striping Volume Selection” on page 106.

Environmental Conditions Affecting Data Set Separation

Under the following environmental conditions, SMS might allocate on the same PCU two data sets that are specified for required or preferred separation:

- Allocation is not SMS-managed.
- Allocation is performed on a system in the SMS complex that runs a prior version of SMS that does not support data set separation.
- During switching or activation of an SMS configuration, the data set separation profile cannot be accessed or fails validation.
- A service that is allocating a temporary data set name does not provide the real data set name to SMS.
- Two data sets that are specified to be separated are allocated at the same time by two different tasks or two different systems. For performance reasons, SMS does not serialize data set allocations.
- A volume is varied online during allocation.
- An IODF change occurs during allocation.
- A data set name that is not listed in the data set separation profile is specified during HSM recover.
- A data set separation profile might have been modified after the configuration was activated. This can result in differences between the data set separation profile data set and the active copy of the data set separation profile.
- SMS does not perform data set separation during RENAME.
- SMS does not perform data set separation during HSM migration to level 1 or level 2 storage.
- SMS *does* perform data set separation during HSM recall.
- SMS does not perform data set separation during full volume image copy.

SMS might not issue the corresponding warning message IGD17372I in those cases where allocation was successful and data set separation was not honored.

Factors Affecting Code Path Length

The following factors affect code path length when data set separation is requested:

- The number of data set names listed in the data set separation profile. (SMS must scan the profile to determine whether data set separation processing is needed for the current allocation.)
- The number of data set names listed within a data set group when data set separation is performed for that group
- The number of volumes that are defined to each PCU for which data set separation processing occurs

Appendix E. Using NaviQuest

DFSMS NaviQuest is a data and storage management tool for implementing, testing, and verifying the DFSMS SMS environment. NaviQuest is installed under the Interactive Storage Management Facility (ISMF) Primary Option Menu and uses the standard Interactive System Productivity Facility (ISPF) panel interface.

With NaviQuest you can:

- Create DFSMS implementation test cases

  Initially, you might use NaviQuest in the design and testing phase of your first automatic class selection (ACS) routines and SMS configuration.
- Run selected DFSMS testing functions in batch
- Update DFSMS configuration values in batch

  NaviQuest lets you run jobs in batch to list, alter, delete or define data class, storage class, management class, storage group or aggregate group. You can also use batch mode to display data class, storage class, management class and aggregate group.
- Create reports interactively or in batch

  NaviQuest reports can be used during your DFSMS planning and design, for both DASD and for tape data migration.
- Perform ongoing storage administration activities

  Use NaviQuest for production assurance prior to any changes to your DFSMS environment. The cross-reference facilities simplify the tasks of testing and verifying SMS configuration changes.

To simplify ongoing activities, the Model Command Generator option, on the Enhanced ACS Management option menu, can be used to create a series of commands or control cards tailored with the data set name or volume serial number that is provided in an ISMF list or DCOLLECT report.

The power and capabilities of DFSMS require that ACS routines assign the correct data class, storage class, management class, and storage group to each data set. Implementing DFSMS, or any change that affects the DFSMS environment, must be tested. The results from the DFSMS configuration are unpredictable if the ACS routines have logic errors or if there are data sets that the routines are not coded to handle.

Before a DFSMS configuration is activated on the production system, you can:

- Easily create test cases to perform extensive testing against test data that represent actual data sets.
- Run the tests in batch, freeing the workstation for other work.
- Compare the test results against the *expected* results.

How to Start NaviQuest

Start NaviQuest by selecting option 11, *Enhanced ACS Management*, from the ISMF Primary Option Menu, as shown in Figure 134 on page 322.

© Copyright IBM Corp. 1986, 2002

```
 Panel  Help
-------------------------------------------------------------------------------
 DGTSMMD2          ISMF PRIMARY OPTION MENU - DFSMS V1 R3
 Enter Selection or Command ===> ____________________________________________

 Select one of the following options and press Enter:

  0  ISMF Profile              - Change ISMF user profile
  1  Data Set                  - Perform Functions Against Data Sets
  2  Volume                    - Perform Functions Against Volumes
  3  Management Class          - Specify Data Set Backup and Migration Criteria
  4  Data Class                - Specify Data Set Allocation Parameters
  5  Storage Class             - Specify Data Set Performance and Availability
  6  Storage Group             - Specify Volume Names and Free Space Thresholds
  7  Automatic Class Selection - Specify ACS Routines and Test Criteria
  8  Control Data Set          - Specify System Names and Default Criteria
  9  Aggregate Group           - Specify Data Set Recovery Parameters
 10  Library Management        - Specify Library and Drive Configuration
 11  Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
  C  Data Collection           - Process Data Collection Function
  L  List                      - Perform functions Against Saved ISMF Lists
  R  Removable Media Manager   - Perform Functions Against Removable Media
  X  Exit                      - Terminate ISMF
 Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 134. ISMF Primary Option Menu for Storage Administrators, DGTSMMD2

Choosing option 11 from the ISMF Primary Option Menu takes you to the NaviQuest Primary Option Menu, as shown in Figure 135.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBSMDP0   ENHANCED ACS MANAGEMENT - NaviQuest PRIMARY OPTION MENU
 Enter Selection or Command ===> ____________________________________________

 Select one of the following options and press Enter:

   1  Test Case Generation
   2  ACS Test Listings Comparison
   3  Enhanced ACS Test Listing
   4  Test Case Update with Test Results
   5  SMS Report Generation
   6  Model Commands Generation
   7  Batch Testing/Configuration Management
   X  Exit









 Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 135. NaviQuest Primary Option Menu, ACBSMDP0

When you select one of the options from the NaviQuest Primary Option Menu, each successive panel will guide you through the choices available for that function.

Terminology

Several new terms are unique to NaviQuest. You will need to familiarize yourself with them before you begin to use this facility.

The following terminology and concepts are used throughout this book:

base line test set
A special case test set, which includes all test cases for data sets not planned for SMS management. Expected values are nulls ('); the associated special subtype prefix is NEVR.

data classification
The process of determining the data types in your installation and identifying data subtypes that require specific DFSMS data services.

data subtype
Groupings of data sets from a data type needing identical DFSMS services, such as performance, backup, migration, or deletion. NaviQuest testing is managed at the subtype level.

data type
Major groupings of data. TSO, test, batch production, online production, system, and temporary are examples of data types that most installations have.

errors
Results that are different from the expected results. Errors are a subset of exceptions.

exceptions
A test case whose results differ from the saved expected results. For a regression test case, this exception is also an error; for an initial test case, this exception might or might not be an error. If the result is equal to the expected value, there is no error.

expected results
Values that you want assigned by the ACS routines for data class, storage class, management class, and storage group for a specific subtype when the ACS routines perform correctly.

initial test
First-time testing of a single data subtype performed prior to converting the data to SMS. The test may include regression tests for other data subtypes that have already been tested successfully.

phase or implementation phase
Data conversion to SMS management of one or more data subtypes with the same data type.

phase test set
Group of subtype test sets that define all data subtypes that make up a phase. Phase test sets are converted to SMS management within a single phase.

regression test
Testing of data subtypes that have already been successfully tested, along with the initial test of the current phase subtype.

results
Values that are assigned by the ACS routines for data class, storage class, management class, and storage group for a specific test case. These results might or might not be correct.

saved expected results
Results that have been saved in the test case after the successful initial test of the subtype test set.

	The results are used by the ACS comparison function during regression testing.
subtype prefix	A unique 1-to-4 character prefix associated with each data subtype. This prefix is used to relate the data classification data subtype to the subtype test set and to group all test cases together for a single subtype test set.
subtype test set	A group of test cases for all data sets associated with the same data subtype. All test cases within one subtype test set have the same expected results.
testbed library	A partitioned data set (PDS) that contains one member to define each test case.
test case	Parameters associated with a single data set that is tested using ISMF option 7.4.3.

Methods for Collecting Test Data

You must become familiar with the methods used to create test data before you begin testing ACS routines.

The procedures for collecting test data are documented in “Testing Procedures” on page 327.

NaviQuest collects data for testing from any of four sources:

- ISMF lists
- DCOLLECT data
- SMF data
- VMA data

Although you can successfully use all four sources to create test data, there are restrictions or considerations associated with choosing one method over another.

ISMF Lists

Using the standard facilities of ISMFsaved lists, you can select the data sets that you want NaviQuest to enter into the testbed library as DFSMS test cases. ISMF filtering capabilities help you to select data associated with one category of data (data subtype) at a time. When ISMF lists are used to create test cases, you can use NaviQuest’s ISMF batch capabilities to generate and save the list.

Consideration: ISMF lists provide greater flexibility. For example, you can tailor the list by removing data sets.

DCOLLECT Data

DCOLLECT data can be used as input to create test cases. The DCOLLECT input is created as an independent batch job by using either the IDCAMS utility program or the ISMF “Data Collection” (option C).

Consideration: The IDCAMS DCOLLECT function runs in batch and is faster than ISMF lists.

SMF Data

Input from system management facility (SMF) data on new allocations can be used to create test cases. This SMF input data is created by using the storage class ACS exit code available in the sample library SYS1.SACBCNTL. Once the exit is installed and a DFSMS minimal or null configuration is running, all new allocations are captured and written to SMF. A program (ACSTST) for postprocessing of the SMF data, which NaviQuest uses to generate standard DFSMS test cases, is also included.

Restriction: Use the SMF data created out of the IGDACSSC exit in selected situations. For example, the extensive testing of system temporary data sets.

VMA Data

VMA data (tape data sets) can be used as input to create test cases. VMA data is gathered in the process of analyzing your tape data. This may be done in preparation for either a tape mount management methodology (TMM) implementation or the design of your system-managed tape environment.

Restriction: Use only VMA data to create test cases for tape data sets.

Data Classification Methodology

Data classification is a methodology for DFSMS implementation that translates high-level requirements into data classes, storage classes, and management classes and identifies the decision points needed in the ACS routines.

Use the data classification technique to build a tree diagram where each major branch (data type) is a collection of data to be migrated to DFSMS management, each in a separate phase of the implementation process.

Figure 136 shows six data types to be managed by SMS.

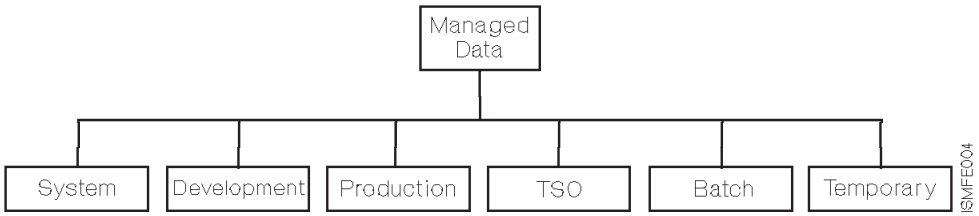


Figure 136. Typical Data Types

After the initial tree diagram has been built, each data type is split further until no more downward splits need to be made. Further splits result in the creation of *data subtypes*. Each data subtype is a distinct set of data sets requiring a unique set of assigned classes. Figure 137 on page 326 is an example of how a TSO data type can be split into different data subtypes.

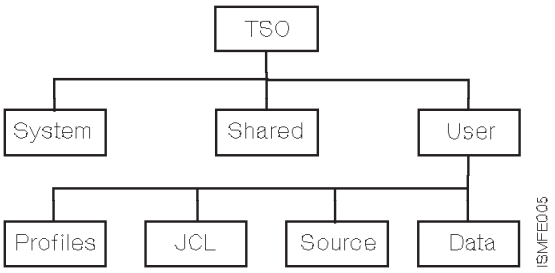


Figure 137. TSO Data Type with Data Subtypes

NaviQuest is designed to support testing of data subtypes determined during data classification. All data sets associated with a single data subtype will have the same expected ACS results. NaviQuest testing at the subtype level lets you manage all test cases for a data subtype and validate the results for that subtype.

Testing ACS Routines

The following is an overview of the procedures required to complete the ACS routine testing phase. Apply this process to each new data subtype until all data has been migrated. For the specific procedure, see “Testing Procedures” on page 327.

- After a sample of the data sets that are to be tested has been chosen (or created), a base line test set is generated, to represent the ACS routines *before* they are changed.
- After the base line test set has been created, the ACS routines are changed (but not activated) to manage the data type. The same set of test cases is then run through ACS testing, generating a new ACS listing.
- A comparison test is run against the base line test set, and the new listing and exceptions are reported in a comparison report. If the change to the ACS routines was made correctly, the exceptions reported should contain only the new test cases.
- If no errors occur after converting the data to SMS management, the test cases are updated to reflect the expected results for future regression testing.
- The current configuration is saved and the new configuration is activated. After the activation, conversion of data to SMS management can proceed.

Setting Up the Test Environment

Prepare your DFSMS testing environment and create the test case library.

1. **Perform data classification to determine the data types.**

   While this step is optional, if you do not perform data classification for the data types, you must still identify the data subtypes that you want NaviQuest to test.

2. **Assign the subtype prefix.**

   If you performed data classification (step 1), you must assign a 1-to-4 character member name prefix to each data classification data subtype. Even if you do not use data classification, you still must identify the different types of tests you want done and assign them a subtype prefix, because each test runs against a different data subtype and produces a different expected result.

   Requirement: For the base line test set, you must use the subtype prefix NEVR.

3. **Determine the amount of testing to be done.**

You can choose to conduct tests of all data sets in a given data subtype or you can conduct tests with a selected percentage of the data sets. If you use all the data sets, the testing is more extensive, with less chance for error. However, you can use a sample of the data to minimize the resources used while testing.

4. **Determine when to place test cases into the testbed library.**

   The testbed library stores test cases for later use in bulk or regression testing.

   You can either initially place test cases for all data subtypes into the testbed library or wait to add each subtype test set to the testbed library at the start of each phase. Fully loading the testbed library with all test cases prior to any testing results in more thorough testing. Alternatively, adding the test cases at the start of each phase reduces the total time taken for test runs.

5. **Determine how the data for the test cases is to be created.**

   You must also decide how you want to collect the data set test cases. For each data subtype, decide if you want ISMF lists, DCOLLECT, VMA, or SMF to be the source of the test cases. See “Methods for Collecting Test Data” on page 324 for an explanation of the data collection methods.

   The percentage of test data and method of collecting test case data can be tailored for each data subtype. This is independent of your choice of storing all test cases as you initially build the testbed library or at the start of each phase’s testing. For example, you may decide to test all production data sets for production data, but to use only a ten percent sample of your test data sets.

6. **Determine the order of testing.**

   Because the order of testing is normally the same as the order of data conversion to SMS management, you must determine the order that you want to test data subtypes with NaviQuest. Only a single data subtype can be initially tested at one time.

   Tip: Other data subtypes that have already been tested are regression tested at this time. NaviQuest testing is usually done against all subtypes of one data type before starting the next.

7. **Allocate the testbed library.**

   You can either create a testbed library or let NaviQuest create one for you. The following instructions will guide you in how to create one.

   To create a testbed library, create a PDS with the following attributes:

   - LRECL = 80
   - BLKSIZE = 0

   Each test case is a member of the PDS. All test cases associated with a single data type have the same subtype prefix and have the identical expected results.

   The amount of space allocated to the library depends on how many data sets the installation has and how many test cases you want to create. For example, on a 3380 device, 1000 test cases take approximately 8 cylinders and 52 directory blocks.

Testing Procedures

When the set up tasks are complete (“Setting Up the Test Environment” on page 326), you can begin NaviQuest testing. The initial testing establishes the base line test set against data sets that will never be DFSMS managed. Because they are not managed, they have an expected result of null (') for each storage class, storage group, data class, and management class.

Requirement: For these test cases, you must use the subtype prefix NEVR.

After the base line test is complete, you can test each phase, or cycle, of the DFSMS implementation, one subtype at a time. Normally, a DFSMS implementation phase is made up of either a single data type or several data subtypes. Each data subtype is tested independently. Once each data subtype tests correctly, you can begin data conversion for DFSMS for that phase.

Recommendation: Put all test cases into the initial base line test set. This will save you from having to repeat adding data types or subtypes one at a time.

Use the following procedure to test your base line test set or any other phase.

1. **Collect data set information for input.**

   With NaviQuest, you can create many test cases at once using input from the following sources:

   - ISMF lists
   - DCOLLECT data
   - SMF data created by a storage class ACS exit
   - VMA data

   The data set test cases must all be representative of the data type that you will migrate to DFSMS management and require the same DFSMS services.

   From the ISMF Primary Option menu, “Data Set” (option 1) offers you two ways to generate data sets samplings: (1) from a saved listing, or (2) from a new listing created from the criteria you specify, such as a VTOC or catalog.

   The multivolume variable is always set to “Yes” in an ISMF table if the data set is not open at the time the table is saved. The value is set correctly at the time the data set is opened, which can sometimes cause errors in the bulk test case generator.

   Recommendations:

   - Generate SMF test cases from the ACSTST program for temporary data sets, because saving tables of temporary data sets might produce errors in bulk test case creation (“Test Case Generation from Saved ISMF List” option 11.1.1).
   - Set the ACQUIRE DATA FROM VOLUME and ACQUIRE DATA IF DFHSM MIGRATED options under the ISMF “Data Set Selection Entry” panel (ISMF option 1) to Y before generating the list.

   After you have created the list, enter the ISMF SAVE command on the command line to save the list into a table. For information on the SAVE command, to *DFSMS: Using ISMF*.

2. **Generate the test cases.**

   Use the “Test Case Generation Selection Menu” panel (option 11.1, shown in Figure 138 on page 329) to turn the ISMF list, DCOLLECT data, SMS data generated by the storage class ACS exit, and VMA data into standard DFSMS test cases.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBSFLG4             TEST CASE GENERATION SELECTION MENU
 Enter Selection or Command ===> ____________________________________________

 Select the input data to be used and press Enter:

   1  Saved ISMF List
   2  DCOLLECT Data
   3  SMF Data
   4  VMA Extract Data









 Use HELP Command for Help; Use END Command to Exit.
```

Figure 138. Test Case Generation Selection Menu, ACBSFLG4

ISMF Lists

To generate test cases from saved ISMF tables, select option 1 (Saved ISMF List) and then use the “Test Case Generation from Saved ISMF List Entry Panel” (option 11.1.1, shown in Figure 139 on page 330). Enter the following information:

- Saved list name previously saved in step 1 on page 328
- Member name prefix (subtype prefix)
- PDS that contains the test cases
- Whether you want to replace the existing test cases with the output test cases

Also select additional values that you want included in the test cases.

Recommendation: If you do not enter a PDS name, NaviQuest will generate one based on the format *userid.Tnn.TESTCASE*. Instead, specify a name so that the test case library conforms to your installation’s naming standards.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDFLG1    TEST CASE GENERATION FROM SAVED ISMF LIST ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate test cases, specify the following information and press Enter:
   Saved ISMF List . . . . . . ________ (Data set list)

   Member Name Prefix  . . . . ____     (1 to 4 alpha characters)
   Test Case PDS . . . . . . . ____________________________________________
     Replace Existing Prefix . _        (Y or N)

 ACS Test Case Variables:
                                                                   More:     +
   Applic  . . . . ________           DD . . . . . . ________
   Def_dataclas  . ________           Def_mgmtclas . ________
   Def_storclas  . ________           Filenum  . . . ____
   Group . . . . . ________           Job  . . . . . ________
   Label . . . . . ___                Libname  . . . ________
   Msvgp . . . . . ________           Pgm  . . . . . ________
   Retpd . . . . . ____               Storgrp  . . . ________

 Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 139. Test Case Generation from Saved ISMF List Entry Panel, ACBDFLG1

DCOLLECT Data

To generate test cases from DCOLLECT data, select option 2 (DCOLLECT Data) and then use “Test Case Generation from DCOLLECT Data Entry Panel” (option 11.1.2, shown in Figure 140).
Enter the following information:

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDFLG5     TEST CASE GENERATION FROM DCOLLECT DATA ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate test cases, specify the following information and press Enter:
   Data Set Containing DCOLLECT Data
     ===> ____________________________________________________

   Member Name Prefix  . . . . ____     (1 to 4 alpha characters)
   Test Case PDS . . . . . . . ____________________________________________
     Replace Existing Prefix . _        (Y or N)
   Number of Test Cases  . . . ____     (1 to 9999, blank=no limit)









 Use HELP Command for Help; Use END Command to Exit.
```

Figure 140. Test Case Generation from DCOLLECT Data Entry Panel, ACBDFLG5

- Data set name
- Number of test cases you want included
- Member name prefix (subtype prefix) of the DCOLLECT test cases

- PDS that contains the test cases
- Whether to replace the existing test cases with the output test cases

Before you can use this function, you must have DCOLLECT data that includes D (data set) records.

SMF Data

To generate test cases from the ACSTST program, select option 3 (SMF Data) and then use “Test Case Generation from SMF Data Entry Panel” (option 11.1.3, shown in Figure 141).
Enter both the data set name containing the system management facility

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDFLG3       TEST CASE GENERATION FROM SMF DATA ENTRY PANEL
 Command ===> ____________________________________________________________

 To generate test cases, specify the following information and press Enter:
   Data Set Containing SMF Data (Type 127 records)
     ===> ____________________________________________________

   Test Case PDS . . . . . . . ____________________________________________


   Note: The IGDACSSC storage class exit must exist and the type 127 SMF
         records must have been extracted.  Program ACBTST and IGDACSSC
          are available in the sample library SYS1.SACBCNTL.


 Use HELP Command for Help; Use END Command to Exit.
```

Figure 141. Test Case Generation from SMF Data Entry Panel, ACBDFLG3

(SMF) data and the name of the test case PDS.

Recommendation: This function requires that you have the IGDACSSC storage class exit installed and have extracted the SMF type 127 records. The ACSTST program is also required. Both are available in the sample library SYS1.SACBCNTL.

VMA Extract Data

To generate a test case from a VMA extract file, select option 4 (VMA Extract Data) and then use the “Test Case Generation from VMA Extract Data Entry Panel” option (11.1.4, shown in Figure 142 on page 332).

```
 Panel  Help
-------------------------------------------------------------------------------
ACBDFLG2   TEST CASE GENERATION FROM VMA EXTRACT DATA ENTRY PANEL
Command ===> ____________________________________________________________________

To generate test cases, specify the following information and press Enter:
  Data Set Containing VMA Extract Data (GFTAXTR output)
    ===> ____________________________________________________

  Member Name Prefix  . . . . ____     (1 to 4 alpha characters)
  Test Case PDS . . . . . . . ____________________________________________
    Replace Existing Prefix . _        (Y or N)
  Number of Test Cases  . . . ____     (1 to 9999, blank=no limit)
  Program Name to Filter on . ________




Note:  Before running this function you must have run GFTAXTR from
       your saved SMF type 14, 15, 21, and type 30 records.


Use HELP Command for Help; Use END Command to Exit.
```

Figure 142. Test Case Generation from VMA Extract Data Entry Panel, ACBDFLG2

Enter the following information:

- Name of the data set containing the VMA Extract data
- Number of test cases you want generated
- Member name prefix (subtype prefix) of the VMA test cases
- Program name, if you want to test the implementation for a particular program

Also include the name of the test case PDS and whether to replace the existing test cases.

Requirement: To use this function, you must have already run GFTAXTR from your saved SMF records (types 14, 15, 21 and 30). JCL for GFTAXTR can be found in SYS1.SAMPLIB member GFTAXTRP.

Batch Options for Creating Test Cases

In addition, you can also use the following batch options to create test cases:

- For option 11.1.1, use the ACBQBAG3 CLIST. See “Generate Test Cases from ISMF-saved Data Set Lists: ACBQBAG3” on page 380.
- For option 11.1.2, use the ACBQBAG1 EXEC. See “Generate Test Cases from DCOLLECT Data: ACBQBAG1” on page 381.
- For option 11.1.3, use the ACBQBAI1 EXEC. See “Generate Test Cases from SMF Data” on page 382.
- For option 11.1.4, use the ACBQBAO3 EXEC. See “Generate Test Cases from VMA Extract Data: ACBQBAO3” on page 383.

Add a 1-to-4 character subtype prefix to each test case member. The prefix must be unique for each data subtype. For example, the first group of TSO data could have subtype prefix TSOA, the second TSOB, and so on.

See step 1 on page 328 for creating the ISMF table. ACBJBAG2, ACBJBAG1, ACBJBAOW, and ACBJBAI1 in SYS1.SACBCNTL JCL library can perform this task in batch (see “How to Run Storage Administration Tasks in Batch” on page 343).

3. **Make ACS routine and construct changes.**

   You must change (but not activate) the ACS code and constructs to reflect the new phase of implementation that you want to test. Before you change the ACS code and construct definitions contained in the source control data set (SCDS), save the old source in case it is needed for recovery.

   For information on recovering the ACDS, refer to “Chapter 12. Recovering Storage Management Subsystem Information” on page 215.

   You can now update the ACS routines to reflect the new data subtype you want migrated to DFSMS.

4. **Update the FILTLISTs**.

   When the ACS code is changed, you might want to use the COPYFILT function of NaviQuest to update all the ACS routines from a common definition of the filter lists. You will be prompted to provide a change log entry that reflects changes you are making to the ACS routines. This entry will be automatically placed into the change log in the ACS routines.

   To use the COPYFILT macro, see “COPYFILT Macro: COPYLIB Facility for FILTLISTs” on page 416.

5. **Translate and validate the ACS routines.**

   You must translate and validate (but not activate) the ACS routines.The ISMF translate function transforms ACS routines into a table format. Translation checks for syntax errors and transforms the ACS routines into a format suitable for input to the validation.

   The ISMF validate function verifies that all possible constructs that can be assigned via the ACS logic have been defined to the SCDS used for testing. ACS routines must be translated before they can be validated; however, validation of ACS routines is optional.

   To translate and validate, you can either use the online ISMF functions or you can use the NaviQuest ISMF-in-batch EXEC.

   For online translation and validation, choose option 7 (ACS Class Selection) from the ISMF Primary Option menu. To translate, choose option 2 (Translate). To validate, choose option 3 (Validate).

   To use the translate facility in batch, see “ACS Routine Translate: ACBQBAO1” on page 372.

   For more information on translation, see “Translating ACS Routines” on page 140. For more information on validation, see “Validating ACS Routines or an Entire SCDS” on page 143.

6. **Run the test cases.**

   Create a **new** ACS listing by using the ISMF “Test ACS Routines” option (7.4.3). The testbed library contains the test cases. Specify an asterisk (*) to run *all* test cases in the library.

   The **new** ACS listing represents the SMS configuration *after* the ACS routines have been changed for the new data subtype.

   Recommendations:

   a. Include the prefix of the subtype tested in the ACS listing data set name, to make it easier to identify which data subtype the listing represents.

   b. Run test cases in batch whenever possible. For sample JCL for testing ACS routines in batch, see “Test ACS Routines: ACBQBAIA” on page 378.

7. **Compare the results of the regression testing.**

   After the base line test, every test includes both testing of new data subtypes and regression testing of previously tested data subtypes, including the base line test set.

   At this time, you use the NaviQuest ACS comparison test function to compare the results of all test cases in the testbed library with their expected results. The ACS comparison test produces a report of exceptions. Since you have not yet stored the expected results of this data subtype's test cases, these test cases appear as exceptions. Later, in step 9 on page 337, you will store the expected results for the current data subtype test cases. But for now, the exceptions you get are either these valid initial (that is, first run) test cases, or they are errors.

   To run the ACS comparison test, choose option 2 from the NaviQuest Primary Option Menu (shown in Figure 143).

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBSMDP0   ENHANCED ACS MANAGEMENT - NaviQuest PRIMARY OPTION MENU
 Enter Selection or Command ===> ____________________________________________

 Select one of the following options and press Enter:

   1  Test Case Generation
   2  ACS Test Listings Comparison
   3  Enhanced ACS Test Listing
   4  Test Case Update with Test Results
   5  SMS Report Generation
   6  Model Commands Generation
   7  Batch Testing/Configuration Management
   X  Exit









 Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 143. NaviQuest Primary Option Menu, ACBSMDP0

On the "ACS Comparison Report" panel, enter the following information:

- Name of your base test case results
- Name of your new test case results
- PDS that contains the test cases
- PDS that contains the exception test cases
- Name of the comparison results data set

```
 Panel  Help
-------------------------------------------------------------------------------
ACBDFLC1           ACS TEST LISTINGS COMPARISON ENTRY PANEL
Command ===> ____________________________________________________________________

To compare ACS listings, specify the following information and press Enter:
  Input Data Sets:
    Base ACS Test Listing (Before latest ACS routine changes)
      ===> ____________________________________________________
    New ACS Test Listing (After latest ACS routine changes)
      ===> ____________________________________________________
  Reference Data Set for Compare:
    Test Case PDS (Test source for listings above)
      ===> ____________________________________________

  Output Data Sets:
    Comparison Results Data Set (Summary of exception test cases)
      ===> ____________________________________________________
      Replace Contents if DSN Exists  . . _ (Y or N)
    Exception Test Case PDS (Contents of exception test cases)
      ===> ____________________________________________
      Replace Contents if DSN Exists  . . _ (Y or N)

Use HELP Command for Help; Use END Command to Exit.
```

Figure 144. ACS Test Listings Comparison Entry Panel, ACBDFLC1

After running “ACS Test Listings Comparison Entry Panel” (option 11.2, shown in Figure 144), you must verify the following items:

- The number of exceptions should be the same as the number of test cases you are currently testing.
- Exceptions should all have the same subtype prefix.
- Each listed test case should have the listed results that you expect.

If changes have been made correctly to the ACS routines, the differences between the two should be only the data subtype that is being initially tested.

Specify a comparison data set name to be used to store the results of the comparison. Also input whether you want to write over the data set specified if it already exists. If N is specified, and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set. Then press the Enter key.

You will be automatically placed into ISPF “browse” when the comparison completes. The comparison data set you are browsing lists only the test cases identified as exceptions.

If exceptions other than the test cases for the subtype you are initially testing are listed, you have probably made an error in coding the revisions to your ACS routines. Changes in coding that have caused errors must be corrected before you can proceed. This means repeating the operations until the test cases match the exceptions.

The following files are created or updated as output:

- Exception PDS
- Comparison data set

Important

Each test is an initial test for one data subtype but may include many regression tests for previously tested data subtypes. Expected values are not stored in the initially tested data subtype until its testing completes successfully.

The ACS comparison performed in step 7 has two functions:

- It validates the regression tests.

  Current test results of each previously tested data subtype should match the saved expected results previously stored with the test cases. If the results are the same, the regression test is successful. If the results differ, there is an error in the new ACS logic; that is, the ACS routine is assigning different values.
- It indicates the subtype test set that is being initially tested.

  Because this is an initial test, this test case has no expected results stored in the test cases, other than null. Thus, during the comparison in step 7 on page 334, all test cases for this new data subtype show an exception; that is, new results will no longer be null.

For more information about running the ACBQBAC1 EXEC in batch, see “ACS Test Listings Comparison: ACBQBAC1” on page 384.

8. **Validate the test results and determine errors.**

   You must manually compare the new test cases to their expected results for the single data subtype that has been initially tested. This comparison determines if there are initial test errors. If the exceptions contain any test cases from the data subtypes previously tested correctly (in regression testing), these exceptions are also errors.

   It is the manual verification of the results that makes sure that the values are the expected results. When all test cases are correct, the test values are stored in the test cases as save expected results, to be used for later regression testing.

   If you find errors, you can generate the NaviQuest ACS cross-reference report for additional information about the specific test cases that produced the errors. Use this report to help you debug the ACS logic. If you find errors (from either step 7 or step 8), you must correct the ACS code before returning to step 3 on page 333 and retest until the data subtype results have no errors.

   If you do not find errors, the test is complete, as all the test cases in the subtype test set have the correct expected results.

   To create an ACS cross-reference report, choose “Enhanced ACS Test Listing Entry Panel” (option 3, shown in Figure 145 on page 337) from the NaviQuest Primary Option Menu (see Figure 135 on page 322). On the “Enhanced ACS Test Listing Entry Panel” fill in the fields with the following names:

   - ISMF test case listing (generated through option 7.4.3)
   - Data set for cross-reference listing (name of data set to contain cross-reference)

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDFLX1            ENHANCED ACS TEST LISTING ENTRY PANEL
 Command ===> _____________________________________________________________________

 To generate enhanced ACS test listing, specify the following and press Enter:
   ACS Test Listing
     ===> ____________________________________________________________

   Enhanced ACS Test Listing
     ===> ____________________________________________________________
     Replace Contents if DSN Exists  . . _ (Y or N)

   Fields to Include in Enhanced ACS Test Listing: (Y or N)

     DSN  . . . . . _               JOBNAME  . . . . _
     EXPDT  . . . . _               SIZE . . . . . . _
     UNIT . . . . . _               PROGRAM  . . . . _




 Use HELP Command for Help; Use END Command to Exit.
```

Figure 145. Enhanced ACS Test Listing Entry Panel, ACBDFLX1

Indicate whether the specified data set should be written over if it already exists. If N is specified, and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.

Specify with a Y or an N which variables you want included in your report. Once you have specified all variables that you want, press the Enter key and the report will be produced.

9. **Save the expected results.**

   Once the subtype test is correct, you can use NaviQuest to place the results of the test (that is, the expected results for later regression testing) into the test case definition as the saved expected results for later regression testing. Test results are only saved after all test cases in the subtype test set have completed with the expected results for that data subtype. The saved expected results will be used for later regression testing, as explained in step 7 on page 334.

   To save the test results, choose “Test Case Update With Test Results Entry Panel” (option 4) from the ISMF Primary Option menu, which takes you to the “Test Case Update With Test Results Entry Panel” (shown in Figure 146 on page 338).

```
 Panel  Help
-------------------------------------------------------------------------------
ACBDFLU1        TEST CASE UPDATE WITH TEST RESULTS ENTRY PANEL
Command ===> ____________________________________________________________________

To update test cases with test results, specify the following and press Enter:

  Input Data Sets:
    New ACS Test Listing (After latest ACS routine changes)
      ===> ____________________________________________________
    Comparison Results Data Set (Summary of exception test cases)
      ===> ____________________________________________________
    Exception Test Case PDS (Contents of exception test cases)
      ===> ____________________________________________

  Input/Output Data Set:
    Test Case PDS
      ===> ____________________________________________

Use HELP Command for Help; Use END Command to Exit.
```

Figure 146. Test Case Update with Test Results Entry Panel, ACBDFLU1

Enter the names of the testbed PDS library, the exception test case PDS, the PDS created in the ACS comparison report, and the new ACS test case listing.

The test case members for the exceptions are read and copied into the testbed library. The saved expected results are obtained from the comparison report and are also saved in the testbed library.

You have now completed testing for this data subtype and can now start testing the next data subtype.

Delete the following data sets at the end of this step:

- Comparison report generated in step 7
- Exception PDS created in step 7
- Base and new ACS listing can be deleted (or printed and deleted)

For more information about running the ACBQBAU1 EXEC in batch, see “Update Test Cases with Expected Results: ACBQBAU1” on page 388.

10. **Test the next data subtype in the current phase.**

    Continue NaviQuest testing for each data subtype in the current DFSMS implementation phase. This testing either repeats Steps 1 through 10 or repeats Steps 3 through 10, depending on whether all subtype test sets are initially placed into the testbed.

    After the initial test of the base line, all additional tests include regression testing along with initial testing.

11. **Activate your new DFSMS configuration.**

    Once an entire phase (that is, all the subtypes within the implementation phase) have tested correctly, you can activate the new configuration by using the SETSMS command at an MVS console.

    For more information on activating your configuration, see “Chapter 10. Activating Storage Management Subsystem Configurations” on page 171.

You might want to use the NaviQuest reporting capabilities to determine the amount of DASD space required to convert the data in each phase, prior to attempting conversion. Use this information to ensure that enough DASD is available for the conversion.

12. **Convert data to SMS management.**

    After activation of your new configuration, you can now migrate the data to DFSMS management. There are several options for doing this data migration:

    - DFDSS COPY
    - DUMP/RESTORE
    - Normal allocation processing
    - MIGRATE/RECALL
    - CONVERTV

NaviQuest Testing Scenario

This customer, initially running with a minimal configuration, plans to convert two data types to SMS management in two phases, one for each data type. The following scenario helps clarify the various testing phases.

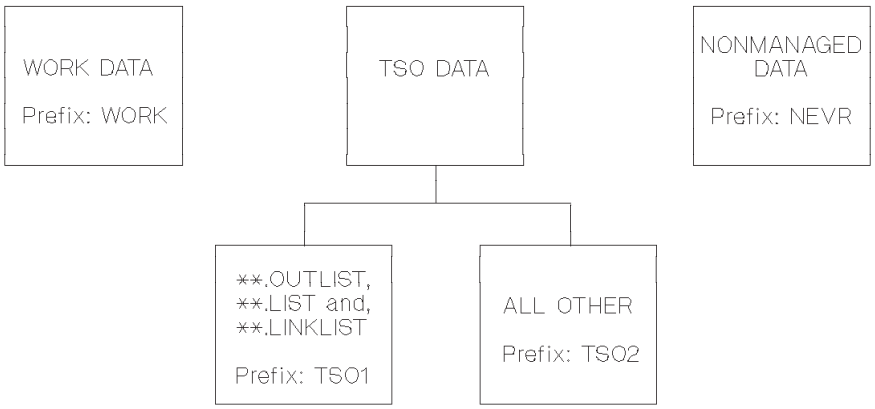
Data Testing

In this example, the customer wants to test two data types with NaviQuest: work data and TSO data. Because these two data types are the initial data types to be tested, establishing and testing non-SMS data must be done first to create the base line test.

The two data types are shown below with typical data classification assignments. Work data is made up of a single data subtype, each work data set having a single expected result from the ACS routines. TSO data is made up of two data subtypes, each subtype having a different set of expected results from the ACS routines.

The subtype prefix assigned for the work data type is WORK. The subtype prefix for the two TSO data subtypes are TSO1 and TSO2.

There is also a data type for all nonmanaged data. This data type is used to create the baseline test cases and must be assigned the subtype prefix NEVR.



Method of Testing

Testing with NaviQuest can be done with a sampling of data sets from each data subtype. In the following testing example, however, only two or three data sets are shown.

All test cases are built *before* any NaviQuest testing begins. As NaviQuest testing begins, there are four prefixes in the test bed library: NEVR, WORK, TSO1, and TSO2, with NEVR representing non-SMS-managed (base line) data sets.

NaviQuest is used to create the test cases from data set lists for each of the four subtype test sets, in this scenario.

Testing Example

Figure 147 on page 341 presents a flowchart of three data types as they progress through the NaviQuest testing phases.

Definitions:

1. The term *systemp* means “system temporary data sets”.These are the WORK data sets.
2. The symbol (") means “null”.
3. Circled numbers represent corresponding steps in the testing procedure.

Refer to Figure 147 on page 341 for the following discussion.

Pass 1 establishes the testing *base line.* Data that is never to be SMS-managed must be included as test cases used in this pass. Additional test cases can also be included; in this example, all planned test data is included in pass 1. The objective of pass 1 is to set the expected results of all base line test cases to *nulls*. The null results establish a set of comparison values so that when future test runs (pass 2) are made, the storage administrator can see where changes occur by identifying the expected results that are no longer nulls.

Pass 2 tests the first set of data to be SMS-managed—in this example, temporary data sets (work data). The storage administrator makes logic changes to the test ACS routines to manage temporary data sets. These changes mean that the expected results for temporary data sets will no longer be null. During pass 2, the NaviQuestcomparison function compares the results with the expected results that were generated and saved in pass 1. If the pass 2 tests are error-free, all the temporary data sets have expected results of nonnulls and they appear as the only exceptions in the comparison report. If pass 2 tests successfully, the storage administrator can save the expected results and begin managing temporary data sets by activating the test ACS routines.

TSO data has two data subtypes: output listings and all other TSO data.Consequently, two passes are needed to complete the TSO testing.

Pass 3 tests the initial part of TSO data to be managed: list output data sets. Again the test ACS routines are changed to now manage both temporary data sets and TSO list output data sets. If pass 3 testing runs error-free, the comparison step has only the test cases for TSO list output. All other test cases have null results or have the expected results of the temporary data sets. If the only exceptions are the TSO list output test cases, pass 3 testing is successful and the expected results are saved. However, the TSO list data is not yet SMS-managed.

Pass 4 tests the rest of the TSO data. The test ACS routines are changed for the additional TSO data. This test *must* complete successfully before any TSO data is converted to SMS management. Again if pass 4 testing runs error free, only the new TSO test cases are identified as exceptions, NaviQuest saves these additional TSO test case expected results, and the storage administrator activates the test ACS configuration. All TSO data can then be converted to SMS management.

Figure 147. Testing Example Flowchart (Part 1 of 2)

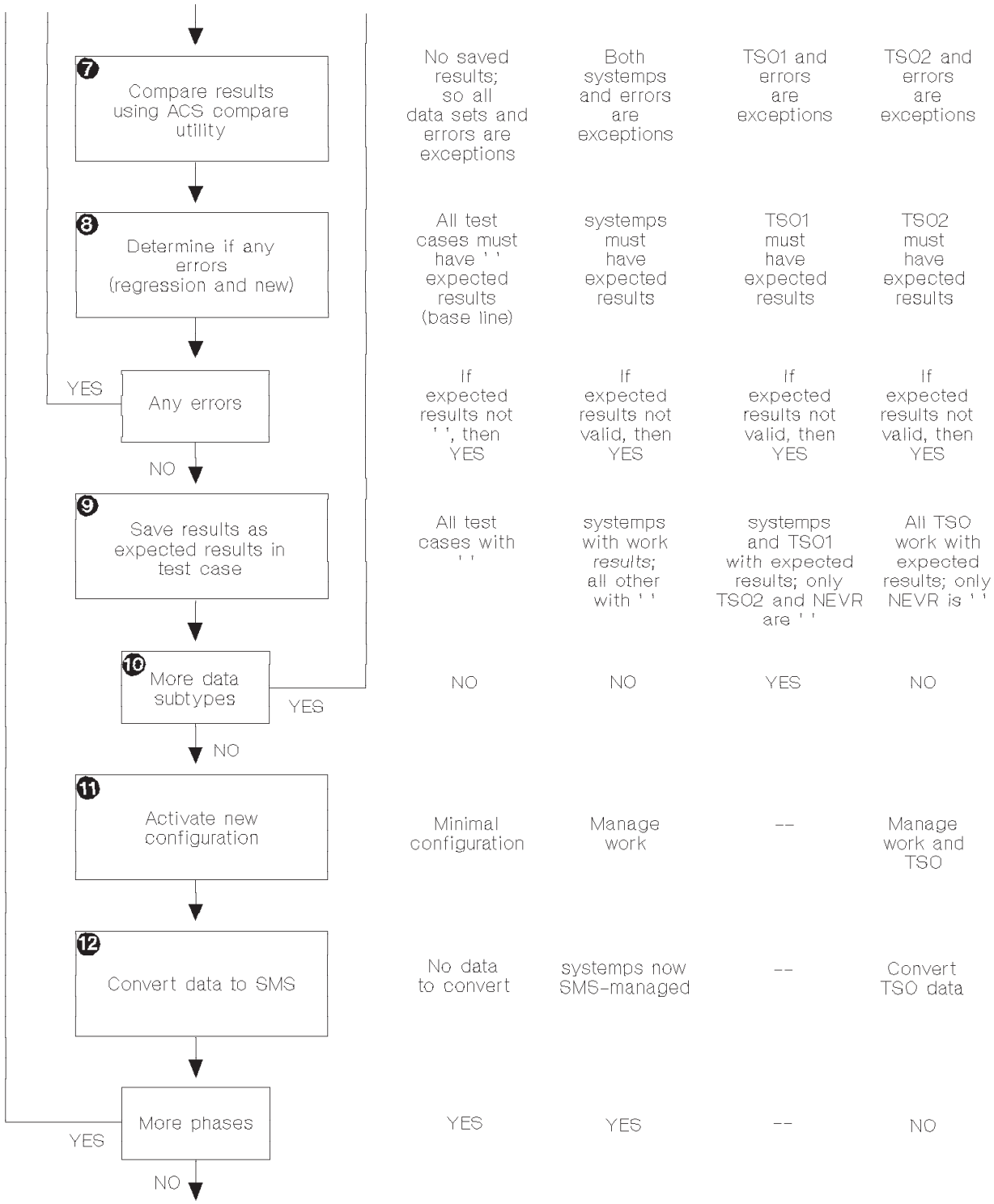


Figure 147. Testing Example Flowchart (Part 2 of 2)

Performing Storage Administration Tasks in Batch

Storage administration tasks that are performed using ISMF options can also be done in batch with JCL, CLISTs, and REXX EXECs that are provided by NaviQuest.

For example:
- Testing the DFSMS configuration
- Performing data set and volume maintenance activities
- Diagnosing data set and volume problems

How to Run Storage Administration Tasks in Batch

To run in batch mode the user is required to:
- Have a TSO user profile
- Know the specific ISMF option that will be run in batch and the task it performs
- Provide the parameters that describe each task. These parameters can be modified by the user.
- Know the required ISPF statements. *Do not modify these ISPF statements.*

ISMF Option 11.7 Batch Testing and Configuration Management

ISMF Option 11.7 (Batch Testing/Configuration Management Selection Menu) lets you select and run batch jobs from an ISMF panel.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBSMDJ1    BATCH TESTING/CONFIGURATION MANAGEMENT SELECTION MENU
 Enter Selection or Command ===> ____________________________________________

 Select an option or enter Data Set to Edit and press Enter:

   1  Saved ISMF List Operations Batch Samples
   2  DCOLLECT Data Operations Batch Samples
   3  Configuration Changes Batch Samples
   4  VMA and SMF Batch Samples









 Data Set to Edit . . ______________________________________________

 Use HELP Command for Help; Use END Command to Exit.
```

Figure 148. Batch Testing/Configuration Management Selection Menu, ACBSMDJ1

1. Select an option and press Enter. You will be advanced to a menu of batch samples.

   Or fill in the “Data Set to Edit” field with the name of the data set containing JCL you want to run. Press Enter. You will be placed in ISPF EDIT mode. Go to step 3.
2. Select one of the batch sample options or fill in the “Data Set to Edit” field with the name of the data set containing JCL you want to run. Press Enter. You will be placed in ISPF EDIT mode.

3. Edit the JCL if needed. While in ISPF EDIT, type SUBMIT at the command line and press ENTER to run the JCL.

   Refer to Sample JCL for Batch, step 2, for information on modifying the JCL samples shipped with NaviQuest.
4. To save the edited JCL, press PF3 or the END command while in ISPF Edit. Fill in the fields and press Enter.

Sample JCL for Batch

NaviQuest provides sample JCL in the SYS1.SACBCNTL library. The JCL can be modified with the parameters for the task that is to be performed. Table 28 on page 345 lists SYS1.SACBCNTL members.

To use the JCL that is provided in SYS1.SACBCNTL, perform the following tasks:

1. Copy the SYS1.SACBCNTL library member that contains the sample JCL for the task that is to be run in batch. See Table 28 on page 345 for a list of members and the tasks they perform.
2. Update the JCL with the appropriate parameters for that task, after DD statement:

   ```
   //SYSTSIN DD *
   ```

   The parameters and their valid values are on the sample JCL.

   Requirements:

   a. Each member contains sample syntax and parameters. Change only the job statement and the syntax and parameters. *Do not change any other JCL.*
   b. ACBJBAOB is called by the other SYS1.SACBCNTL library members during batch processing. *Do not modify ACBJBAOB when it is called by the other JCL members of SYS1.SACBCNTL.*

      Figure 149 on page 347 shows the JCL in ACBJBAOB.

ISPSTART Batch Parameters for NaviQuest

The following ISPSTART batch parameters have been coded with the appropriate values for NaviQuest. Refer to the *ISPF V4R2 Dialog Developer's Guide and Reference* for additional information about these parameters.

BATSCRD Screen depth

BATSCRW Screen width

BDISPMAX Maximum number of panel displays for a session

BDISPMAX represents the total number of panel display calls. This value is coded to avoid loops.

BREDISPMAX

Maximum number of times the same panel can be displayed

The job will terminate if this limit is reached. This value is coded to avoid loops.

SYS1.SACBCNTL Sample JCL Library

Table 28 lists the sample job library members:

Table 28. SYS1.SACBCNTL Sample JCL Library Member List

Member	Function
ACBJBAA1	Aggregate group define/alter/display
ACBJBAB1	Base configuration define/alter/display
ACBJBAC1	ACS test listings comparison report
ACBJBAC2	Translate ACS routines, validate SCDS, test ACS routines, and generate ACS comparison report
ACBJBAD1	Data class define/alter/display
ACBJBAG1	Generate test cases from previously collected DCOLLECT data (‘D’ records)
ACBJBAG2	Generate test cases from a previously saved table (data set list)
ACBJBAI1	Generate test cases using data from SMF type 127 records.
ACBJBAI2	Generate data set list and save it in a table
ACBJBAI4	Generate DASD volume list, save it in a table, and save the query
ACBJBAI5	Generate DASD volume list and save it in a table
ACBJBAI7	Generate data set list, save it in a table, and save the query
ACBJBAI8	Generate DASD volume list, save it in a table, and generate a report from it
ACBJBAI9	Generate DASD volume list using a previously saved query and save it in a table
ACBJBAIA	Generate ISMF mountable tape volume list, save it in a table, and generate report from it
ACBJBAIB	Alter storage group volume status
ACBJBAIC	Test ACS routines
ACBJBAID	Generate mountable tape volume list and save it in a table
ACBJBAIH	Generate data set list using a previously saved query and save it in a table
ACBJBAII	Generate Aggregate Group list and report
ACBJBAIJ	Generate Aggregate Group list and save it
ACBJBAIK	Generate Aggregate Group report from a saved list
ACBJBAIL	Generate Data Class list and report
ACBJBAIM	Generate Data Class list and save it
ACBJBAIN	Generate Data Class report from a saved list
ACBJBAIO	Generate Management Class list and report
ACBJBAIP	Generate Management Class list and save it
ACBJBAIQ	Generate Management Class report from a saved list
ACBJBAIR	Generate mountable optical volume list and save it
ACBJBAIS	Generate mountable optical volume list and report
ACBJBAIT	Generate mountable optical volume report from a saved list
ACBJBAIU	Generate Storage Class list and report
ACBJBAIV	Generate Storage Class list and save it
ACBJBAIW	Generate Storage Class report from a saved list
ACBJBAIX	Generate Storage Group list and report
ACBJBAIY	Generate Storage Group list and save it

Table 28. SYS1.SACBCNTL Sample JCL Library Member List (continued)

Member	Function
ACBJBAIZ	Generate Storage Group report from a saved list
ACBJBAJ1	Management class define/alter/display
ACBJBAJ2	Pool storage group define/alter
ACBJBAJ3	Tape storage group define/alter
ACBJBAJ7	Object backup storage group define/alter
ACBJBAJ8	VIO storage group define/alter
ACBJBAJA	Dummy storage group define/alter
ACBJBAJB	Object storage group define/alter
ACBJBAJC	Management Class Delete
ACBJBAJD	Data Class Delete
ACBJBAJE	Storage Class Delete
ACBJBAJF	Storage Group Delete
ACBJBAJG	Aggregate Group Delete
ACBJBAM1	Generate model command from saved ISMF table (data set list)
ACBJBAM2	Model commands from DCOLLECT data
ACBJBAOB	Library allocation
ACBJBAOD	Generate data set report from a previously saved table (data set list)
ACBJBAOF	Generate volume report from a previously saved table (DASD volume list)
ACBJBAOQ	Translate ACS Routines
ACBJBAOS	Validate SCDS
ACBJBAOT	Generate tape report from a previously saved table (tape volume list)
ACBJBAOU	Generate data set list, save it in a table, and generate report from it
ACBJBAOW	Generate test cases from VMA extract file
ACBJBAO7	Generate DCOLLECT output
ACBJBAR2	Generate SMS configuration report From DCOLLECT data
ACBJBARD	Generate data set report from DCOLLECT data
ACBJBAS1	Storage Class define/alter/display
ACBJBAU2	Update test cases test cases with expected results
ACBJBAU4	Job to customize the job card for all NaviQuest jobs
ACBJBAXV	Generate DASD volume report from DCOLLECT data
ACBJBAX1	ACS cross reference Report

ACBJBAOB JCL

Figure 149 shows the JCL for ACBJBAOB.

```
//ACBJBAOB PROC CLIST1='SYS1.DGTCLIB',
//*-------------------------------------------------------------------*/
//* CLIST1 SHOULD BE THE FILE #1 FROM THE INSTALLATION TAPE          */
//* NOTE THAT THIS IS A FB CLIST LIBRARY; IF YOUR INSTALLATION       */
//* USES VB CLIST LIBRARIES, YOU MUST CONVERT THEM YOURSELF          */
//*-------------------------------------------------------------------*/
//             PLIB1='SYS1.DGTPLIB',
//*-------------------------------------------------------------------*/
//* PLIB1 SHOULD BE THE FILE #3 FROM THE INSTALLATION TAPE           */
//*-------------------------------------------------------------------*/
//             LOAD1='SYS1.DGTLLIB',
//*-------------------------------------------------------------------*/
//* LOAD1 SHOULD BE THE FILE #5 FROM THE INSTALLATION TAPE           */
//*-------------------------------------------------------------------*/
//             MLIB1='SYS1.DGTMLIB',
//*-------------------------------------------------------------------*/
//* MLIB1 SHOULD BE THE FILE #6 FROM THE INSTALLATION TAPE           */
//*-------------------------------------------------------------------*/
//             TABL2='userid.TEST.ISPTABL'
//*-------------------------------------------------------------------*/
//* TABL2 IS THE DATA SET FOR SAVING ISMF TABLES; YOU SHOULD         */
//* ALLOCATE THIS DATA SET WITH THE SAME DCB PARAMETERS AS THE       */
//* ISMF DGTTLIB DATA SET; TABLE CAN BE LARGE - ALLOCATE A LARGE     */
//* DATA SET.  Be sure you change 'userid' to YOUR userid.           */
//*-------------------------------------------------------------------*/
//*******************************************************************/
//*$MAC(ACBJBAOB) COMP(5695DF123): BATCH  - CALLED PROC             */
//*                                                                 */
//* PROPRIETARY V3 STATEMENT                                        */
//* LICENSED MATERIALS - PROPERTY OF IBM                            */
//* 5647-A01                                                        */
//* (C) COPYRIGHT 1995, 2000  IBM CORP.                             */
//* END PROPRIETARY V3 STATEMENT                                    */
//*                                                                 */
```

Figure 149. SYS1.SACBCNTL member ACBJBAOB JCL for Batch Functions (Part 1 of 2)

```
//* CHANGE ACTIVITY:                                                    */
//*$K0=NAVIQUEST,HACS110,95/08/04,SNJADR: INITIAL VERSION        @K0A*/
//*$K1=KNQ0003,HACS110,95/08/04,SNJTCS: COMMENT CORRECTION       @K1C*/
//*02/29/96  TCS - HACS110:  REMOVED SEQUENCE NUMBERS        @WA18945*/
//*$L0=NAVIQUEST,HACS120,96/06/18,SNJTCS: RELEASE 2 CHANGES      @L0A*/
//*$L1=KN20035,HACS120,96/07/12,SNJTCS: TYPO CORRECTED           @L0A*/
//*10/14/96  CCY - HDZ11C0: Ship as SPE                      @WA22861*/
//*06/13/97  CCY - HDZ11D0: Change IBMUSER to 'userid'       @WA27246*/
//*$T1=NQ5,HDZ11E0,061997,SNJTCS: ISP V3R5M0 Datasets deleted    @T1A*/
//*08/27/98  TCS - HDZ11D0: BLKSIZE=0  fix                   @WA34895*/
//*11/23/98  TCS - HDZ11E0: Add missing comma                @WA36007*/
//*01/21/99  TCS - HDZ11E0: Changing UNIT from 3380 to SYSDA @WA36897*/
//*                                                                   */
//*******************************************************************/
//*******************************************************************/
//* PROC STEP STEP1 - INVOKES IKJEFT01                                */
//*******************************************************************/
//STEP1    EXEC PGM=IKJEFT01,DYNAMNBR=50,REGION=(6144K),TIME=(300)
//*-----------------------------------------------------------------*/
//STEPLIB  DD DSN=&LOAD1,DISP=SHR
//*-----------------------------------------------------------------*/
//* IN ISPPLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST PANELS     */
//*-----------------------------------------------------------------*/
//ISPPLIB  DD DSN=&PLIB1,DISP=SHR,BLKSIZE=0
//         DD DSN=ISP.SISPPENU,DISP=SHR            /* ISPF PANELS    */
//*        DD DSN=SYS1.DGTPLIB,DISP=SHR            /* ISMF PANELS    */
//*-----------------------------------------------------------------*/
//* IN ISPMLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST MESSAGES   */
//*-----------------------------------------------------------------*/
//ISPMLIB  DD DSN=&MLIB1,DISP=SHR,BLKSIZE=0
//         DD DSN=ISP.SISPMENU,DISP=SHR            /* ISPF MESSAGES  */
//*        DD DSN=SYS1.DGTMLIB,DISP=SHR            /* ISMF MESSAGES  */
//*-----------------------------------------------------------------*/
//* IN ISPSLIB, INCLUDE THE ISPF, PDF AND ISMF SKELETONS             */
//*-----------------------------------------------------------------*/
//ISPSLIB  DD DSN=ISP.SISPSENU,DISP=SHR,           /* ISPF SKELETONS */
//          BLKSIZE=0
//         DD DSN=SYS1.DGTSLIB,DISP=SHR            /* ISMF SKELETONS */
//*-----------------------------------------------------------------*/
//* IN ISPTLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST TABLES     */
//*-----------------------------------------------------------------*/
//ISPTLIB  DD DSN=&&TEMP,DISP=NEW,UNIT=SYSDA,BLKSIZE=0,
//            SPACE=(TRK,(1,1,1)),DCB=(ISP.SISPTENU)
//         DD DSN=&TABL2,DISP=SHR
//         DD DSN=ISP.SISPTENU,DISP=SHR            /* ISPF TABLES    */
//         DD DSN=SYS1.DGTTLIB,DISP=SHR            /* ISMF TABLES    */
//*-----------------------------------------------------------------*/
//* IN SYSPROC, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST CLISTS     */
//*-----------------------------------------------------------------*/
//SYSPROC  DD DSN=&CLIST1,DISP=SHR,BLKSIZE=0
//         DD DSN=ISP.SISPCLIB,DISP=SHR            /* ISPF CLISTS    */
//*        DD DSN=SYS1.DGTCLIB,DISP=SHR            /* ISMF CLISTS    */
//*-----------------------------------------------------------------*/
//ISPTABL  DD DSN=&TABL2,DISP=SHR,BLKSIZE=0
//SYSPRINT DD SYSOUT=(,)
//SYSUDUMP DD SYSOUT=*
//SYSTSPRT DD SYSOUT=(,),OUTLIM=20000
//ISPLOG   DD SYSOUT=(,),DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
//ISPPROF  DD DSN=&&PROF,DISP=(NEW,DELETE,DELETE),
//         DCB=(ISP.SISPTENU),SPACE=(TRK,(1,1,1)),UNIT=SYSDA
//         PEND
```

Figure 149. SYS1.SACBCNTL member ACBJBAOB JCL for Batch Functions (Part 2 of 2)

NaviQuest CLISTS and REXX EXECS

The CLISTs and REXX EXECs that are called by the sample JCL shipped with NaviQuest, can also be called by JCL that *you* code, to perform storage administration tasks. Your JCL must call the appropriate CLIST or REXX EXEC for the task that is to be performed. Use Table 29 to find the CLIST or REXX EXEC you want to use, then update the ISPSTART statement with its name and parameters.

Tip: The naming convention for data set names in the ISPSTART command in the sample jobs is the same as in ISMF. Quoted names are fully qualified and unquoted names will have the PREFIX added as a high-level qualifier.

Refer to the JCL sample jobs in SYS1.SACBCNTL to see the complete syntax for each task.

Recommendations:

- Do not modify these REXX EXECs and CLISTs.
- The REXX EXECs and CLISTs do not create a listing; they create an ISMF-saved table (except the test ACS routines task), which is similar to running ISMF interactively and then issuing a SAVE *'xxxxxxxx'* command. Save the table, then use the ISMF 11.5.*n* options to produce a flat file of the table for printing.

Table 29. CLISTs and REXX EXECs for Storage Administration Tasks

Interactive ISMF Option	Storage Administration Task	CLIST or REXX EXEC
1	Generate data set list	ACBQBAI2
2.1	Generate DASD or optical device volume list	ACBQBAI4
2.2	Generate optical volume list	ACBQBAIE
2.3	Generate tape volume list	ACBQBAI6
3.2	Display management class	ACBQBAJ1
3.3	Define management class	ACBQBAJ1
3.4	Alter management class	ACBQBAJ1
4.2	Display data class	ACBQBAD1
4.3	Define data class	ACBQBAD1
4.4	Alter data class	ACBQBAD1
5.2	Display storage class	ACBQBAS1
5.3	Define storage class	ACBQBAS1
5.4	Alter storage class	ACBQBAS1
6.2	Define dummy storage group Define object backup storage group Define object storage group Define pool storage group Define tape storage group Define VIO storage group	ACBQBAJA ACBQBAJ7 ACBQBAJB ACBQBAJ2 ACBQBAJ3 ACBQBAJ8

Table 29. CLISTs and REXX EXECs for Storage Administration Tasks (continued)

Interactive ISMF Option	Storage Administration Task	CLIST or REXX EXEC
6.3	Alter dummy storage group Alter object backup storage group Alter object storage group Alter pool storage group Alter tape storage group Alter VIO storage group	ACBQBAJA ACBQBAJ7 ACBQBAJB ACBQBAJ2 ACBQBAJ3 ACBQBAJ8
6.4	Storage group add/delete volume	ACBQBAI9
7.2	ACS routine translate	ACBQBAO1
7.3	SCDS validation	ACBQBAO2
7.4.3	Test ACS Routines	ACBQBAIA
8.1	Display base configuration	ACBQBAB1
8.2	Define base configuration	ACBQBAB1
8.3	Alter base configuration	ACBQBAB1
9.2	Display aggregate group	ACBQBAA1
9.3	Define aggregate group	ACBQBAA1
9.4	Alter aggregate group	ACBQBAA1
11.1.1	Test cases from ISMF-saved list	ACBQBAG3
11.1.2	Test cases from DCOLLECT data	ACBQBAG1
11.1.3	Test cases from SMF data	ACSTST program
11.1.4	Test cases from VMA extract data	ACBQBAO3
11.2	ACS test listings comparison	ACBQBAC1
11.3	Enhanced ACS test listing	ACBQBAX1
11.4	Update test cases with expected test results	ACBQBAU1
11.5.1	Data set report from ISMF-saved list	ACBQBAR1
11.5.2	DASD volume report from ISMF-saved list	ACBQVAR1
11.5.3	Tape volume report from ISMF-saved list	ACBQBAR4
11.5.4	Data set report from DCOLLECT data	ACBQBAR7
11.5.5	DASD volume report from DCOLLECT data	ACBQBAR6
11.5.6	SMS configuration report from DCOLLECT data	ACBQBAR8
11.6.1	Model command from ISMF-saved list	ACBQBAM1
11.6.2	Model command from DCOLLECT data	ACBQBAM2

Generate a Data Set List: ACBQBAI2

ACBQBAI2 is called by the following SYS1.SACBCNTL members to generate the data set list in batch:

ACBJBAI2 Generate data set list and save it in a table
ACBJBAI7 Generate data set list, save it in a table, and save the query
ACBJBAOU Generate data set list, save it in a table, and generate report

See Figure 150 on page 354 for the sample JCL and parameters.

Save or Delete the Data Set List: Use the following command to specify whether or not the ISMF table should be saved after it has been generated:

ISPSTART CMD(ACBQBAI2 SAVE|DELETE *tablnm parameters* ...)

SAVE *tablnm* Indicates that the generated data set list is to be saved in data set *tablnm* with the specified parameters.

DELETE Indicates that the specified data sets in the generated list are to be deleted.

Rule: Use the DELETE option carefully. Before you use this option, generate the table (with the SAVE option), print it, and then examine the table to see which data sets are to be deleted.

Use the parameters in Table 30 with ACBQBAI2. At least one OP and one value should be included when specifying a parameter.

Table 30. Using ACBQBA12 Parameters

Parameters	Description
ALLOCSP(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' is EQ, NE, GT, LE, GE, LT, 'nnn1' is the allocate space value, in kilobytes (KB); BOOL is AND or OR; OP2 has the same values as OP1; and 'nnn2' has the same values as nnn1
ALLOCUT(OP dd1 dd2 dd3 dd4)	Where OP is EQ or NE; allocation unit, BLK/TRK/ABS/CYL/MB/KB/BYT
BLKSIZE(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GT, LE, or NE; 'nnn1' and 'nnn2' are the block size values; BOOL is AND or OR
BLKUNUSED(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are blocks unused values; and BOOL is AND or OR
CATNAME(catalog name)	Where 'catalog name' is the name of the catalog to be searched for the dsns
CATVOL(DDDDDD)	Where 'DDDDDD' is the volume serial. This is the variable used when you are generating a data set list from the catalog and you want to limit the data sets generated to those on a particular volume
CCSIDDSC(OP dd1 dd2 dd3 dd4)	Where OP is EQ or NE; CCSID description 0 to 65534 or a 1 to 17 character string that is not a number 0-65534
CFCSTNM(OP dd1 dd2 dd3 dd4)	Where OP is EQ or NE; CF cache structure name
CFCTNM(OP dd1 dd2 dd3 dd4)	CF cache set name

Table 30. Using ACBQBA12 Parameters (continued)

CHGIND (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO for change indicator bit setting for the DSN
CFMOST(OP dd1 dd2 dd3 dd4)	Where 'OP' is EQ or NE; CF monitor status ON/OFF
CFSTIND(OP dd1 dd2 dd3 dd4)	Where 'OP' is EQ or NE; CF status indicator CE/CQ/VRRQ/VRLS/VQ
COMPFMT (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO for compressed format for the data set
CREATEDT (OP1 dat1 BOOL OP2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR
DATACLS(OP DDD1 DDD2 DDD3 DDD4)	Where 'OP' is EQ or NE, and 'DDD1' through DDD4' are the data classes
DDMATTR(OP dd1 dd2 dd3 dd4)	Where 'OP' is EQ or NE; DDM attributes YES/NO
DEVTYPE(OP DDD1 DDD2 ... DDD8)	Where 'OP' is EQ, NE, GT, LE, and so on, and 'DDD1' through DDD8 are the device types
DSENV(OP dd1 dd2 dd3 dd4)	Where 'OP' is EQ or NE; dataset environment MANAGED/UNMANAGED
DSN(list of dsns)	Where ″list of dsns″ is the format used by ISMF for generating lists of data sets and volumes. The default is '**'
DSNTYP(OP dd1 dd2 dd3 dd4)	Where 'OP' is EQ or NE; DS name type EXTENDED/HFS/LIBRARY/OTHERS
DSORG(OP DD1 DD2 ... DD8)	Where 'OP' is EQ or NE, and 'DD1' through 'DD8' are the data set organizations
ENTRYTYP(OP DDD1 DDD2 ... DDD12)	Where 'OP' is EQ or NE, and 'DDD1' through 'DDD12' are the entry types (for example, DEFERRED, AIX, CLUSTER, and GDG)
EXPIREDT (OP1 dat1 BOOL OP2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR
HSMDATA(Y|N)	This specifies whether or not the user wants to use the ACQUIRE DATA FROM DFHSM option. The default is 'N'.
LASTBKUP (OP1 dat1 BOOL O2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR
LASTREF (OP1 dat1 BOOL OP2 dat2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'dat1' and 'dat2' are dates in the yyyy/mm/dd format; BOOL is AND or OR
LRECL(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' is EQ, GE, GT, LE, LT, or NE, and 'nnn1' is the lrecl specification; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1
LISTTYP(I|E)	Inclusive/Exclusive or I/E Inclusive: Display list by Inclusive criteria Exclusive: Display list by Enclusive criteria
MGMTCLS(OP nnnnnn1 ... Nnnnnn4)	Where 'OP' is EQ or NE, and 'nnnnnn1'through 'nnnnnn4' are the management classes
MULTVOL (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO if data set is multivolume

Table 30. Using ACBQBA12 Parameters (continued)

NOTUSED%(OP1 nn1 BOOL OP2 nnn2)	Where 'OP1' is EQ, NE, GT, LE, and 'nn1' is the % of space not used; BOOL is AND or OR; OP2 has the same values as OP1; and 'nnn2' has the same values as nnn1
NUMEXT(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' is EQ, NE, GT, LE, and 'nnn1' is the extent specification; BOOL is AND or OR; OP2 has the same values as OP1; and 'nnn2' has the same values as nnn1
NUMSTRIPE (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are stripe number values; and BOOL is AND or OR
OPTIMAL (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are optimal blksize values; and BOOL is AND or OR.
OWNER (OP DD1 DD2 DD3 DD4)	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are owners of the the datasets.
QSAVE(nnnnnnnn)	Where 'nnnnnn' is the query name to be created with all saved variables
QUERY(nnnnnnnn)	Where 'nnnnnn' is the query name to be used for all the variables
REBLOCK(OP DDD1 DDD2 ... DDD3)	Where 'OP' is EQ and and 'DDD1' through 'DDD3' are either YES or NO.
RECFMT(OP DDD1 DDD2 .. DDD8)	Where 'OP' is EQ or NE, and 'DDD1' through 'DDD8' are the record formats. List built from the specified criteria.
SECALLOC (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are secondary allocation values; and BOOL is AND or OR.
SOURCEGL(1|2)	Specifies whether the generated list comes from a saved list or is a new generated list. The default is '2'. When generating a list from the VTOC (SOURCENL is 1), you must specify VTOCVSER. The following parameters will be ignored: CATNAME, CATVOL, VTOCDATA, and HSMDATA. When generating a list from the catalog (SOURCENL is 2), use CATVOL to for the volume serial. Specify VTOCDATA and HSMDATA if needed. VTOCVSER is ignored.
SOURCENL(1|2)	Specifies whether the generated list comes from 1 - VTOC , or 2 - Catalog. The default is '2'.
STORCLS(OP nnnnnn1 .. Nnnnnn4)	Where 'OP' is EQ or NE, and 'nnnnnn1' through 'nnnnnn4' are the storage classes
USEDSPC(OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' is EQ, NE, GT, LE, and 'nnn1' is the amount of used space in KB; BOOL is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1
USERDATAREDUCT% (OP1 nnn1 BOOL OP2 nnn2)	Where 'OP1' and 'OP2' are EQ, GE, GT, LE, LT, or NE; 'nnn1' and 'nnn2' are% of user data reduction; and BOOL is AND or OR
VTOCDATA(Y|N)	Specifies whether the user desires the ACQUIRE DATA FROM VOLUME OPTION. The default is 'N'.
VTOCVSER(VVVVVV)	Specifies the volsers whose VTOCs are to be searched. The user may specify from 1–6 alphanumeric characters and an asterisk for filtering.

Sample JCL for Generating a Data Set List:

```
//*******************************************************************/
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO CREATE ISMF DATA SET LIST IN BATCH AND SAVE IT    *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*     CHANGE JOBCARD                                              *
//*     CHANGE PREFIX                                               *
//*     CHANGE PARAMETERS                                           *
//*                                                                 *
//*   PARAMETERS:                                                   *
//*                                                                 *
//*                                                   46A/C @WA28670*
//*     PARAMETER FOLLOWING SAVE - NAME OF THE SAVED LIST (OUTPUT)  *
//*     ALLOCSP - 0 to 9999999 (in Kilo Bytes)                      *
//*     ALLOCUT - Allocation Unit, BLK/TRK/ABS/CYL/MB/KB/BYT     @U1A*
//*     BLKSIZE - 0 to 99999 (in Bytes)                             *
//*     BLKUNUSED - 0 to 9999999 (in Kilo Bytes)                    *
//*     CATNAME - ICF CATALOG name                                  *
//*     CATVOL  - VOLUME used in catalog list                       *
//*     CHGIND  - YES/NO                                            *
//*     COMPFMT - YES/NO                                            *
//*     CREATEDT - YYYY/MM/DD  (1900/01/01 to 2155/12/31)           *
//*                YY/MM/DD ( (19)00/01/01 to (19)99/12/31 )        *
//*     CCSIDDSC - CCSID Description                            3@U1A*
//*                0 to 65534 or A 1 to 17 character string         *
//*                that is not a number 0-65534                     *
//*     CFCSTNM  - CF Cache Structure name                       @U1A*
//*     CFCTNM   - CF Cache Set name                             @U1A*
//*     CFMOST   - CF Monitor status ON/OFF                      @U1A*
//*     CFSTIND  - CF status indicator CE/CQ/VRRQ/VRLS/VQ        @U1A*
//*     DATACLS - DATA CLASS name                                    *
//*     DDMATTR - DDM Attributes YES/NO                          @U1A*
//*     DEVTYPE - 3380/3390/9345 for DASD types                      *
//*             - 3480/3480X/3490/3590-1 for TAPE types              *
//*     DSENV   - Dataset Environment MANAGED/UNMANAGED          @U1A*
//*     DSNTYP  - DS name type EXTENDED/HFS/LIBRARY/OTHERS       @U1A*
//*     DSN     - DATA SET NAME                                      *
//*               ('**' requires catalog name)                       *
//*     DSORG   - DA/DAU/IS/ISU/PO/POU/PS/PSU/UN/VS                  *
//*     ENTRYTYP - AIX/ALIAS/CATALOG/DATA/INDEX/NONVSAM/PATH/        *
//*                DEFERRED/GDG/GDS/ROLLOFF                          *
//*     EXPIREDT - YYYY/MM/DD (1900/01/01 to 2155/12/31)         or  *
//*              - YY/MM/DD ( (19)00/01/01 TO (19)99/12/31 )     or  *
//*              - NEVER    or     1999/00/00                        *
//*     HSMDATA  - Y / N  (Catalog List)                             *
//*     LASTBKUP - YYYY/MM/DD  (1900/01/01 to 2155/12/31)            *
//*                YY/MM/DD ( (19)00/01/01 to (19)99/12/31 )         *
//*     LASTREF  - YYYY/MM/DD  (1900/01/01 to 2155/12/31)            *
//*                YY/MM/DD ( (19)00/01/01 to (19)99/12/31 )         *
//*     LISTTYP  - Inclusive/Exclusive or I/E                   3@U1A*
//*                Inclusive: Display list by Inclusive criteria     *
//*                Exclusive: Display list by Enclusive criteria     *
```

Figure 150. Sample JCL for ACBJBAI2 (Part 1 of 2)

```
//*     LRECL   - 0 to 99999 (in Bytes)                              *
//*     MGMTCLS - MANAGEMENT CLASS name                              *
//*     MULTVOL - YES/NO                                             *
//*     NOTUSED% - 0 to 100                                          *
//*     NUMEXT   - 0 to 999                                          *
//*     NUMSTRIPE - 1 to 16                                          *
//*     OPTIMAL - 0 to 99999 (in Bytes)                              *
//*     OWNER   - OWNER of the datasets                              *
//*     QSAVE   - QUERY Name to be saved                             *
//*     QUERY   - QUERY Name to be used                              *
//*     REBLOCK - YES/NO                                             *
//*     RECFMT  - A/B/D/F/M/FS/VS/T/U/V                              *
//*     SECALLOC - 0 to 99999999 (in Kilo Bytes)                     *
//*     SOURCEGL - 1 / 2 (2-new list)                                *
//*     SOURCENL - 1 / 2 (1-VTOC 2-catalog)                          *
//*     STORCLS - STORAGE CLASS name                                 *
//*     USEDSPC - 0 to 9999999 (in Kilo Bytes)                       *
//*     USERDATAREDUCT% - 0 to 99                                    *
//*     VTOCDATA - Y / N (Catalog List)                              *
//*     VTOCVSER - VTOC VOLUME SERIAL NUMBERS (VTOC list)            *
//*                                                                  *
//********************************************************************
//SAVELIST EXEC ACBJBAOB,
//         PLIB1=SYS1.DGTPLIB,
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAI2 SAVE DSNLIST +
VTOCDATA(Y) HSMDATA(Y) +
ALLOCSP(GT 500) DSORG(NE PS) +
DSN('IBMUSER.**')) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//********************************************************************
```

Figure 150. Sample JCL for ACBJBAI2 (Part 2 of 2)

Generate a DASD Volume List: ACBQBAI4

ACBQBAI4 is called by the following SYS1.SACBCNTL members to generate the list of DASD volumes in batch:

ACBJBAI4 Generate a DASD volume list, save it in a table, and save the query

ACBJBAI5 Generate a DASD volume list and save it in a table

ACBJBAI8 Generate a DASD volume list, save it in a table, and generate a report from it

For the sample JCL and parameters, see Figure 151 on page 358.

For ACBQBAI4, use the parameters listed in Table 31. At least one OP and one value should be included when specifying a parameter.

Table 31. ACBQBAI4 Parameters

Parameter	Description
ALLOCSP(OP1 nnn1 BOOL OP2 nnn2)	Specifies the amount of allocated space. Accepted values for nnn1 are from 0 to 9999999 (in KB); 'OP1' is EQ, NE, GT, LE, GE and LT; 'BOOL' is AND or OR; 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1.
CDSNAME(cds.name)	The SCDS that extracts the volume information from for the volume list
CFVOLST(OP nn1 nn2 nn3 nn4)	'OP' is EQ or NE; CF volume status ENABLED/QUIESCING/QUIESCED
CFWSTAT(OP nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with specified cached-write status. Accepted values for 'nnn#' are non-E, ACTIVE, INACTIVE, PENDING, or PINNED; 'OP' is EQ or NE.
DEVTYPE(xxxxxxxx)	Device type to use for the volume list. If you specify the DEVTYPE parameter, NaviQuest attempts to determine the device type of the volume (for example, 3380-K). Currently NaviQuest only determines the correct device type if the volume is an entire MVS volume, that is, the using volume is not a VM minidisk. If the type cannot be determined, the default value xxxx-?, where *xxxx* is the generic device type.
DFWSTAT(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with specified DASD fast-write status. Accepted values for 'nnn#' are non-E, ACTIVE, INACTIVE, PENDING, or PINNED; 'OP' is EQ or NE
DUPLXSTAT(OP1 nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with specified duplex status. Accepted values for 'nnn#' are non-E, SIMPLEX, PRIMARY, SECONDARY, PRI-PEN, SEC-PEN, PRI-SUS, and SEC-SUS; 'OP' is EQ or NE
FRAG(OP1 nnn1 BOOL OP2 nnn2)	Specifies the fragmentation index. Accepted values for 'nnn1' are from 0 to 999; 'OP1' is EQ, NE, GT, LE, GE and LT; 'BOOL' is AND and OR; 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1
FREEDSCB(OP1 nnn1 BOOL OP2 nnn2)	Specifies the number of free DSCBs. Accepted values for 'nnn1' are from 0 to 99999;. 'OP1' is EQ, NE, GT, LE, LT and GE; 'BOOL' is AND or OR; 'OP2' 'has the same values as OP1, and 'nnn2' has the same values as nnn1.

Table 31. ACBQBAI4 Parameters (continued)

FREESPC(OP1 nnn1 BOOL OP2 nnn2)	Specifies the amount of free space. Accepted values for 'nnn1' are from 0 to 9999999 (in KB); 'OP1' is EQ, NE, GT, LE, LT and GE; 'BOOL' is AND and OR; 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1.
FREESPC%(OP1 nnn1 BOOL OP2 nnn2)	Limits the volumes included in the list to those with specified proportions of free space. Accepted values for 'nnn1' are from 0 to 100, with no % sign; 'OP1' is EQ, NE, GT, LE, LT and GE; 'BOOL' is AND or OR; OP2 has the same values as OP1, and 'nnn2' has the same values as nnn1.
FREEVIR(OP1 nnn1 BOOL OP2 nnn2)	Specifies the amount of free VIRs. Accepted values for 'nnn1' are from 0 to 99999; 'OP1' is EQ, NE, GT, LE, LT and GE;'BOOL' is AND or OR; 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1.
FREEXT(OP1 nnn1 BOOL OP2 nnn2)	Specifies the number of free extents. Accepted values for 'nnn1' are from 0 to 99999; 'OP1' is EQ, NE, GT, LE, LT and GE; 'BOOL' is AND or OR; 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1.
FROMDEV(nnn)	First device number in the range to be listed in the volume list. Used with the LASTDEV parameter.
INDEX(OP nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with acceptable VTOC status. Accepted values for 'nnn#' are ENABLED, DISABLED, or non-E; 'OP' is EQ or NE.
LASTDEV(nnn)	Last device number in the range to be listed in the volume list. Used with the FROMDEV parameter.
LISTTYP(I|E)	Inclusive/Exclusive or I/E Inclusive: Display list by Inclusive criteria Exclusive: Display list by Enclusive criteria
LRGEXT(OP1 nnn1 BOOL OP2 nnn2)	Limits the volumes in the list to those with largest extents of a specified size, in KB. Accepted values for 'nnn1' are from 1 to 9999999; 'OP1' is EQ, NE, GT, LE, LT and GE; 'BOOL' is AND or OR, 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1.
OTHERDEV(OP1 nnn1 BOOL OP2 nnn2)	Specifies the address of the duplex copy secondary device. Accepted values for 'nnn1' are from 0 to FFF; 'OP1' is EQ, NE, GT, LE, LT and GE; 'BOOL' is AND or OR; 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1.
PHYDATA(Y|N)	Specifies whether or not the user desires the ACQUIRE PHYSICAL DATA OPTION. The default is 'N'.
PHYSTAT(OP nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with a specific SMS status. Accepted values for 'nnn#' are INITIAL, CONVERT, non-SMS, or UNKNOWN; 'OP' is EQ or NE.
QSAVE (nnnnnnnn)	Where 'nnnnnnnn' is the query name to be created with all saved variables.
QUERY (nnnnnnnn)	Where 'nnnnnnnn' is the query name to be used for all variables.
RDCACHE(OP1 nnn1 nnn2 nnn3 Nnn4)	Limits the volumes included in the list to those with a specified caching status. Accepted values for 'nnn#' are non-E, ACTIVE, INACTIVE, or PENDING; 'OP' is EQ or NE.
SHRDASD(OP nnn1 nnn2 nnn3 Nnn4)	Limits the volumes included in the list to those with DASD that either are or are not shareable between multiple CPUs. Accepted values for 'nnn#' are YES or NO; 'OP' is EQ or NE.
SOURCEGL(1|2)	Specifies whether the generated list come from (1) a saved list or (2) is a new list built from the the specified criteria. The default is '2'.

Table 31. ACBQBAI4 Parameters (continued)

SOURCENL(1|2)	Specifies whether the generated list comes from PHYSICAL(1) or DFSMS(2). The default is '1'. When generating an SMS volume list (SOURCENL is '2'), STORGRP defaults to '*' and CDSNAME defaults to 'ACTIVE' if no values are specified.
SPCDATA(Y|N)	Specifies whether or not the user desires the ACQUIRE SPACE DATA OPTION. The default is 'N'.
STORGRP(XXXXXXXX)	This is the STORGRP to extract the volume information from.
SUBSYSID(OP1 nnn1 BOOL OP2 nnn2)	Limits the volumes included in the list to those with specified subsystems. You may specify a single subsystem number or a range of subsystem numbers. Accepted values for 'nnn1' are from 0001 to 00FF; 'OP1' is EQ, NE, GT, LE, LT and GE; 'BOOL' is AND or OR; 'OP2' has the same values as OP1, and 'nnn2' has the same values as nnn1.
USEATTR(OP nnn1 nnn2 nnn3 nnn4)	Limits the volumes included in the list to those with allowable DASD attributes. Accepted values for 'nnn#' are PUB, PRIV, or STOR; 'OP' is EQ or NE.
VOL (list of VOLS)	Where ″list of VOLS″ is the format used by ISMF for generating lists of data sets and volumes. The default is '*'.
VOLSTYPE(1|2|3)	Specifies whether the generated list will come from ONLINE(1), NOT ONLINE(2), or EITHER(3). The default is '1'.

Sample JCL for Generating a Volume List*:*

```
//*********************************************************************
//*                                                                   *
//* SAMPLE JCL TO GENERATE ISMF DASD VOLUME LIST IN BATCH, SAVE IT,   *
//* AND SAVE THE QUERY ALSO                                           *
//*                                                                   *
//*   INSTRUCTIONS BEFORE SUBMITTING:                                 *
//*                                                                   *
//*      CHANGE JOBCARD                                               *
//*      CHANGE PREFIX                                                *
//*      CHANGE PARAMETERS                                            *
//*                                                                   *
//*   PARAMETERS:                                                     *
//*                                                 36A/C @WA29014 *
//*   PARAMETER FOLLOWING SAVE - NAME OF SAVED LIST (OUTPUT)          *
//*   ALLOCSP   - 0 to 9999999 (in Kilo Bytes)                        *
//*   CDSNAME   - Control Dataset Name                                *
//*   CFVOLST   - CF Volume status ENABLED/QUIESCING/QUIESCED    @U1A*
//*   CFWSTAT   - ACTIVE/INACTIVE/NONE/PENDING/PINNED                 *
//*   DEVTYPE   - 3380/3390/9345                                      *
//*             - Installation defined esoteric names(like SYSDA ..) *
//*   DFWSTAT   - ACTIVE/INACTIVE/NONE/PENDING/PINNED                 *
//*   DUPLXSTAT - NONE / PPRI-FAI / PPRI-PEN / PPRI-SUS / PPRIMARY/   *
//*               PRI-PEN / PRI-SUS / PRIMARY / PSEC-FAI / PSEC-PEN/ *
//*               PSEC-SUS /PSECNDRY/ SEC-PEN  / SEC-SUS / SECONDRY/ *
//*               SIMPLEX / SPAR-BRK / SPAR-PEN / SPARE               *
```

Figure 151. Sample JCL for ACBJBAI4 (Part 1 of 2)

```
//*   FRAG      - 1 to 999                                          *
//*   FREEDSCB  - 0 to 99999                                        *
//*   FREESPC   - 0 to 9999999 (in Kilo Bytes)                      *
//*   FREESPC%  - 0 to 100 (with no % sign)                         *
//*   FREEVIR   - 0 to 99999                                        *
//*   FREEXT    - 0 to 99999                                        *
//*   FROMDEV   - 1 to 4 HexaDecimal digits                         *
//*   INDEX     - DISABLED/ENABLED/NONE                             *
//*   LASTDEV   - 1 to 4 HexaDecimal digits (should be >= FROMDEV)   *
//*   LISTTYP   - Inclusive/Exclusive or I/E                    3@U1A*
//*               Inclusive: Display list by Inclusive criteria     *
//*               Exclusive: Display list by Enclusive criteria     *
//*   LRGEXT    - 1 to 9999999 (in Kilo Bytes)                      *
//*   OTHERDEV  - 1 to 4 Hex Decimal digits                         *
//*   PHYDATA   - Y / N                                             *
//*   PHYSTAT   - CONVERT/INITIAL/NONSMS/UNKNOWN                    *
//*   QSAVE     - Query Name to be saved                            *
//*   QUERY     - Query Name to be used                             *
//*   RDCACHE   - ACTIVE/INACTIVE/NONE/PENDING                      *
//*   SHRDASD   - YES / NO                                          *
//*   SOURCEGL  - 1 / 2 (2-New List)                                *
//*   SOURCENL  - 1 / 2 (1-Physical, 2-SMS)                         *
//*   SPCDATA   - Y / N                                             *
//*   STORGRP   - Stotage Group name (SMS only)                     *
//*   SUBSYSID  - 0001 to FFFF (in HexaDecimal)                     *
//*   USEATTR   - PRIV/PUB/STOR                                     *
//*   VOL       - Volume serial                                     *
//*   VOLSTYPE  - 1 / 2 / 3 (1-On Line 2-Not On Line 3-Either)      *
//*                                                                 *
//*******************************************************************
//DASDLST EXEC ACBJBAOB,
//        PLIB1=SYS1.DGTPLIB,
//        TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER) MSGID
ISPSTART CMD(ACBQBAI4 +
SAVE CHKFRAG QSAVE(CHKFRAG) +
SPCDATA(Y) PHYDATA(Y) +
VOL(TSC*) FRAG(GT 450)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
```

Figure 151. Sample JCL for ACBJBAI4 (Part 2 of 2)

Generate a Tape Volume List: ACBQBAI6

ACBQBAI6 is called by the following SYS1.SACBCNTL members to list the mountable tape volumes in batch:

ACBJBAID Creates ISMF table of scratch tapes in a library, and then prints the table.

ACBJBAIA Creates ISMF table of all tapes in a library.

For the sample JCL and parameters for ACBJBAIA, see Figure 152 on page 361.

For ACBQBAID, use the parameters listed in Table 32.

Table 32. ACBQBAID Parameters

Parameter	Description
LIBNAME (nnn)	Specifies the library name to query against. The default is '*'.
SOURCEGL(1|2)	Specifies whether the generated list comes from (1) a saved list or (2) a new list built from the specified criteria. The default is '2'.
STORGRP(XXXXXXXX)	Specifies the storage group to extract the volume information from. The default is '*'.
VOL(list of VOLS)	Where 'list of VOLS' is in the format for ISMF. The default is '*'.

Sample JCL for Generating a Tape List:

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE AN ISMF MOUNTABLE TAPE VOLUME LIST,      *
//* SAVE IT, AND GENERATE A TAPE VOLUME REPORT FROM IT              *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PROFILE PREFIX COMMAND                              *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* STEP TO GENERATE TAPE VOLUME LIST AND SAVE IT                   *
//*                                                                 *
//*   VOL     - TAPE VOLUMES TO BE SELECTED                         *
//*   LIBNAME - LIBRARY NAMES TO BE SELECTED                        *
//*   STORGRP - STORAGE GROUPS TO BE SELECTED                       *
//*   PARAMETER FOLLOWING SAVE - NAME FOR THE SAVED LIST (OUTPUT)   *
//*                                                                 *
//*******************************************************************
//GENLIST  EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
DELETE TAPEVOL.REPORT
ISPSTART CMD(ACBQBAI6 SAVE TAPELIST +
VOL(*) LIBNAME(L*) STORGRP(*)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
//*                                                                 *
//* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED   *
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY    *
//*       DATA SET.                                                 *
//*                                                                 *
//*******************************************************************
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD  DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),
//    BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN  DD *
/*
//*******************************************************************
//*                                                                 *
//* STEP TO GENERATE THE TAPE VOLUME REPORT FROM THE SAVED LIST     *
//*                                                                 *
//*   PARAMETER FOR ACBQBAR4 - SAVED LIST NAME (FROM GENLIST STEP)  *
//*   ISPFILE - TAPE VOLUME REPORT (OUTPUT, FROM ALCISPFL STEP)     *
//*   SYSIN - KEY WORDS SPECIFYING COLUMNS TO BE INCLUDED IN THE    *
//*           GENERATED REPORT                                      *
//*                                                                 *
//*******************************************************************
```

Figure 152. Sample JCL for ACBJBAIA (Part 1 of 2)

```
//GENREP  EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR4 TAPELIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//ISPFILE DD  DSN=IBMUSER.TAPEVOL.REPORT,DISP=OLD
//SYSIN DD  *
VOLSER
USEATTR
VOLERROR
CKPT
LIBNAME
STORGRP
MEDIA
RECTECH
COMPTYPE
LASTWRITE
LASTMOUNT
LASTEJECT
EXPDTVOL
CRTDTVOL
VOLLOC
TITLE=STATUS OF TAPES AS ON 06/01/96
/*
//*******************************************************************
//*                                                                 *
//* COPY THE REPORT                                                 *
//*                                                                 *
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)                           *
//*   SYSUT2 - OUTPUT                                               *
//*                                                                 *
//*******************************************************************
//TAPGEN EXEC PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.TAPEVOL.REPORT,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*******************************************************************
```

Figure 152. Sample JCL for ACBJBAIA (Part 2 of 2)

Define/Alter/Display Management Class: ACBQBAJ1

SYS1.SACBCNTL member ACBJBAJ1 calls ACBQBAJ1 to define, alter, or display data set management class in batch.

See Figure 153 for the sample JCL and parameters.

Sample JCL for Define/Alter/Display Management Class:

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO DEFINE/ALTER/DISPLAY MANAGEMENT CLASSES IN BATCH  *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*     CHANGE JOBCARD                                              *
//*     CHANGE PREFIX                                               *
//*     CHANGE PARAMETERS                                           *
//*                                                                 *
//*   PARAMETER FOLLOWING ACBQBAJ1 - DEFINE or ALTER or DISPLAY     *
//* ********************** ADD BEG ***********************************
//*  (Define or Alter)                                              *
//*                                                                 *
//* Required Fields:                                     2@WA49380  *
//*                                                                 *
//*   SCDS - SCDS in which MANAGEMENT CLASS is to be DEF/ALT/DISP   *
//*                                                                 *
//*   MGMTCLAS - MANAGEMENT CLASS to be DEFINED/ALTERED/DISPLAYED   *
//*                                                                 *
//* Optional Fields:                                       2@WA49380 *
//*                                                                 *
//*   DESCR    : Type in  remarks  about the MGMTCLAS which is being *
//*              defined/altered, not  exceeding  120  chars.       *
//*                                                                 *
//*   EXPNOUSE : The datasets  will  expire if they are not used for *
//*              the number of days specified here.                 *
//*                                                                 *
//*              Possible values   1 - 9999, NOLIMIT. If  NOLIMIT is *
//*              specified the DS would not expire.                 *
//*              Valid  only if retention  period or expiration date *
//*              is not specified by the end user or is not derived  *
//*              from the data class.                               *
//*                                                                 *
//*   EXPDTDY  : Datasets expires after  DATE/DAYS entered here.    *
//*              Possible values 0 - 9999,YYYY/MM/DD or NOLIMIT.    *
//*                                                                 *
//*   RETNLIM  : Possible values 0 - 9999, NOLIMIT.                 *
//*              Use this field to control what a user or Data class *
//*              can specify for retention period or expiration date *
//*              during allocation. The affect of the values entered *
//*              in this field are explained below.                 *
//*                                                                 *
//*              0         ->  Do not  use  the RETPD and EXPDT that *
//*                            the user or Dataclass specified.     *
//*              1 - 9999  ->  Use  this  value only if the RETPD or *
//*                            EXPDT is more than this limit.       *
//*              NOLIMIT   ->  Do not set a limit to RETPD or EXPDT. *
//*                                                                 *
```

Figure 153. Sample JCL for ACBJBAJ1 (Part 1 of 6)

```
//*   PARTREL  : Possible values  Y, C, YI, CI or N .                *
//*              Use this field (PARTIAL RELEASE) to specify whether *
//*              allocated but unused space can be  released for DSs *
//*              in this MGMTCLS. This one applies  only to VSAM DSs *
//*              in extended format or NON-VSAM datasets. The values *
//*              entered would have following results.               *
//*                                                                  *
//*              Y  -> Release unused space automatically during the *
//*                    Space Management cycle.                       *
//*                                                                  *
//*              C  -> Unused  space  can  be released automatically *
//*                    only if a secondary allocation exists for the *
//*                    dataset.                                      *
//*                                                                  *
//*              YI -> Release unused space when a dataset is closed *
//*                    or during the Space Management cycle,whichever*
//*                    comes first.                                  *
//*                                                                  *
//*              CI -> Unused  space  for  data sets  with secondary *
//*                    allocation is released either when a data set *
//*                    is  closed  or  during  the  Space Management *
//*                    cycle, whichever comes first.                 *
//*                                                                  *
//*              N  -> Do not release unused space.                  *
//*                                                                  *
//*   PRINOUSE : Use this  field to specify when to migrate the DSs  *
//*              in this class. The possible values are              *
//*                                                                  *
//*              0         ->  To Migrate data sets  as  soon as the *
//*                            space management function of DFSMShsm *
//*                            is run and data integrity age is met. *
//*                                                                  *
//*              1 to 9999 ->  Migrate  data  sets  out  of  primary *
//*                            storage if  they have been unused for *
//*                            this number of days or longer.        *
//*                                                                  *
//*              BLANK     ->                                        *
//*                                                                  *
//*   LVINOUSE : Use this filed to  specify whether DSs  can migrate *
//*              to  LEVEL 1 storage  and  how  long they can remain *
//*              there. The possible values are,                     *
//*                                                                  *
//*              0         -> No  migration to Level 1. DSs  migrate *
//*                           directly from primary storage to LVL 2 *
//*                                                                  *
//*              1 to 9999 -> The  total  number of consecutive days *
//*                           that datasets  must  remain unaccessed *
//*                           before  becoming  eligible  to migrate *
//*                           from LVL 1 to LVL 2.                   *
//*                                                                  *
//*              NOLIMIT   -> Datasets  can  not  migrate to LEVEL 2 *
//*                           automatically, and remain in LVL 1 for *
//*                           an unlimited period.                   *
//*                                                                  *
//*              BLANK     ->                                        *
//*                                                                  *
//*   CMDORAUT : If  migration is allowed, this field determines how *
//*              the  migration is initiated. Possible values are,   *
//*                                                                  *
//*              BOTH     -> DSs  can migrate either automatically   *
//*                          or by command.                          *
//*              COMMAND  -> Data sets can migrate by command only.  *
//*                                                                  *
//*              NONE     -> Data sets cannot migrate.               *
//*                                                                  *
```

Figure 153. Sample JCL for ACBJBAJ1 (Part 2 of 6)

```
//*   PRIGDGEL : Valid  for  Generation  Data Group (GDG) DSs  only. *
//*              This  field  specifies  how  many  of  the  newest  *
//*              generations  of a GDG are to have  normal priority. *
//*              Possible  values  are  0 - 255 or blank.For Example *
//*              enter    100  if  you  want GDG generations  older  *
//*              than the  most recent 100, to  migrate before  non  *
//*              generation datasets.                                *
//*                                                                  *
//*   GDGROLL  : This field  specifies  whether the  Generation  DSs *
//*              in  this  MGMTCLS will expire or migrate after they *
//*              have  been  removed  from  the  GDG. The  possible  *
//*              values are, MIGRATE,EXPIRE or blank.                *
//*                                                                  *
//*   BACKUPFR : This  field  specifies the  backup frequency.   The *
//*              possible values are,                                *
//*                                                                  *
//*              0        -> Backup each dataset only when the volume*
//*                          it resides on is backed up.             *
//*              1 - 9999 -> If dataset is  changed  in the interval *
//*                          between  backups, extend  the  interval *
//*                          for atleast this many number of days.   *
//*              BLANK    ->                                         *
//*                                                                  *
//*   NUMBKDSE : Maximum  number  of  Backups  that   can   be  kept *
//*              concurrently. Possible values are, 1 - 100,BLANK.   *
//*                                                                  *
//*   NUMBKDSD : Specifies  the  maximum no of Backups to keep after *
//*              the dataset is deleted. Possible values are         *
//*                                                                  *
//*              0       -> All backups that were created are erased *
//*                         after the dataset is deleted.            *
//*              1 - 100 -> The maximum no. of backups to keep after *
//*                         a dataset has been deleted.              *
//*              BLANK    ->                                         *
//*                                                                  *
//*   RETDYDSD : Specifies how long a most recent  backup version of *
//*              a deleted dataset will be kept. Possible values are *
//*                                                                  *
//*              1 - 9999 -> After a dataset is deleted keep its most*
//*                          recent  backup  version  for these many *
//*                          days.                                   *
//*                                                                  *
//*              NOLIMIT  -> The  backup  version  will  be kept for *
//*                          unlimited period.                       *
//*                                                                  *
//*              BLANK    ->                                         *
//*                                                                  *
//*   RETDYEXT : Specifies the retention period  for  a dataset that *
//*              pre-date  the  most recent  backup. Possible values *
//*              are,                                                *
//*                                                                  *
//*              1 - 9999 -> Each backup version of a  dataset other *
//*                          than the  recent copy will be  kept for *
//*                          these many days.                        *
//*                                                                  *
//*              NOLIMIT  -> All  backup  versions will  be kept for *
//*                          unlimited period.                       *
//*                                                                  *
//*              BLANK    ->                                         *
//*                                                                  *
```

Figure 153. Sample JCL for ACBJBAJ1 (Part 3 of 6)

```
//*   CMDBKUP  : Specifies who will have authority to perform command*
//*              backups. Possible values are,                       *
//*                                                                  *
//*              ADMIN  -> Only Storage Administrator ,              *
//*                                                                  *
//*              BOTH   -> Both Storage Administrator and end users. *
//*                                                                  *
//*              NONE   -> Neither end user nor Storage Administrator*
//*                                                                  *
//*   AUTOBKUP : Specifies whether the datasets in this  MGMTCLS are *
//*              eligible for automatic backup. Possible  values are *
//*              Y -> Yes , N -> No                                  *
//*                                                                  *
//*   BKUPTECH : Specifies  BACKUP COPY TECHNIQUE to be used.        *
//*              Possible values are,                                *
//*                                                                  *
//*              R  -> Concurrent copy technique must be used.       *
//*              P  -> Concurrent copy technique should be used.     *
//*              S  -> With out the concurrent copy technique.       *
//*                                                                  *
//*   TMSCYRS  : No of years  that must pass since the creation date *
//*              before class transition occurs. Possible values are *
//*              0 - 9999, or BLANK.                                 *
//*                                                                  *
//*   TMSCMTH  : No of months that must pass since the creation date *
//*              before class transition occurs. Possible values are *
//*              0 - 9999, or BLANK.                                 *
//*                                                                  *
//*   TMSCDYS  : No of days   that must pass since the creation date *
//*              before class transition occurs. Possible values are *
//*              0 - 9999, or BLANK.                                 *
//*                                                                  *
//*   TMSLUYRS : No of years that must pass since the last reference *
//*              date before class transition occurs.Possible values *
//*              are 0 - 9999, or BLANK.                             *
//*                                                                  *
//*   TMSLUMTH : No of months that must pass since the last reference*
//*              date before class transition occurs.Possible values *
//*              are 0 - 9999, or BLANK.                             *
//*                                                                  *
//*   TMSLUDYS : No of days  that must pass since the last reference *
//*              date before class transition occurs.Possible values *
//*              are 0 - 9999, or BLANK.                             *
//*                                                                  *
//*   PMTHODAY : The day of the month that class transition occurs.  *
//*              Possible values, 1 - 31, FIRST, LAST or BLANK       *
//*                                                                  *
//*   PQUAODAY : The day of the each quarter the CT occurs.          *
//*              Possible values, 1 - 92, FIRST, LAST or BLANK       *
//*                                                                  *
//*   PQUAIMTH : Month of each quarter the CT occurs.                *
//*              Possible values, 1 - 3, or BLANK                    *
//*                                                                  *
//*   PYRLODAY : The day of each year the CT occurs.                 *
//*              Possible values, 1 - 366,FIRST,LAST or BLANK        *
//*                                                                  *
//*   PYRLIMTH : The month of each year the CT occurs.               *
//*              Possible values, 1 - 12, or BLANK                   *
//*                                                                  *
```

Figure 153. Sample JCL for ACBJBAJ1 (Part 4 of 6)

```
//*   VERSIONS : Specify  how  many versions of an  aggregate  group *
//*              associated  with  the  management  class  are to be *
//*              maintained. Possible values are 1 - 9999, NOLIMT or *
//*              BLANK. If BLANK is specified no aggregate group BKP *
//*              is maintained.                                      *
//*                                                                  *
//*   RTNOVERS : Specify  how  long the only version of an aggregate *
//*              group is kept. Possible values are 1 - 9999,NOLIMIT *
//*              or BLANK.                                           *
//*                                                                  *
//*   RTOVUNIT : Specify  the  unit of measure for the length of time*
//*              specified in the above field. Possible  values  are *
//*              D -> Days, W -> Weeks, M -> Months, Y -> Years  and *
//*              BLANK.                                              *
//*                                                                  *
//*   RTNEVERS : Specify the  time  periods for which backup versions*
//*              of an aggregate group are to be kept.Possible values*
//*              are 1 - 9999,NOLIMIT and BLANK.                     *
//*                                                                  *
//*   RTEVUNIT : Specify the unit  of measure for the length of time *
//*              specified in the above field. Possible  values  are *
//*              D -> Days, W -> Weeks, M -> Months, Y -> Years  and *
//*              BLANK.                                              *
//*                                                                  *
//*   CPYSERLN : Specifies  whether you want  processing of a backup *
//*              copy of an aggregate group to continue if a  shared *
//*              enqueue cannot be obtained for the  datasets  being *
//*              backed up. Possible values are,                     *
//*              C -> Continue, F -> Fail or BLANK.                  *
//*                                                                  *
//*   ACPYTECH : Specifies ABACKUP COPY TECHNIQUE to be used.        *
//*              Possible values are,                                *
//*                                                                  *
//*              R  -> Concurrent copy technique must be used.       *
//*              P  -> Concurrent copy technique should be used.     *
//*              S  -> With out the concurrent copy technique.       *
//*                                                                  *
//*   UPDHLVLSCDS: When modifying an SCDS, that was formatted with a *
//*                higher level of SMS, using a lower level of SMS   *
//*                will make this application fail unless you        *
//*                specify the UPDHLVLSCDS parameter as 'Y'.         *
//*                Default is 'N'.                           @WA41441*
//*                                                                  *
//*                Possible values : Y/N/BLANK               @WA41441*
//************************ ADD END ***********************************
//********************************************************************
//*
//*   STEP1 - SET UP PARAMETERS
//*
//********************************************************************
//STEP1   EXEC ACBJBAOB,
//        TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//TEMPFILE  DD  DSN=&&TEMPFILE,DISP=(NEW,PASS),
//  SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ1 DEFINE/ALTER +
SCDS(TEMP.SCDS) +
MGMTCLAS() +
DESCR() +
EXPNOUSE() +
EXPDTDY() +
```

Figure 153. Sample JCL for ACBJBAJ1 (Part 5 of 6)

```
RETNLIM() +
PARTREL() +
PRINOUSE() +
LVINOUSE() +
CMDORAUT() +
PRIGDGEL() +
GDGROLL() +
BACKUPFR() +
NUMBKDSE() +
NUMBKDSD() +
RETDYDSD() +
RETDYEXT() +
CMDBKUP() +
AUTOBKUP() +
BKUPTECH() +
TMSCYRS() +
TMSCMTH() +
TMSCDYS() +
TMSLUYRS() +
TMSLUMTH() +
TMSLUDYS() +
PMTHODAY() +
PQUAODAY() +
PQUAIMTH() +
PYRLODAY() +
PYRLIMTH() +
VERSIONS() +
RTNOVERS() +
RTOVUNIT() +
RTNEVERS() +
RTEVUNIT() +
CPYSERLN() +
ACPYTECH() +
UPDHLVLSCDS() +
)
/*
//*******************************************************************
//*
//*   STEP2 - EXECUTE THE DEFINE/ALTER
//*
//*******************************************************************
//STEP2   EXEC ACBJBAOB
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD DSN=&&TEMPFILE,DISP=(OLD,DELETE,DELETE)
//*******************************************************************
//*
//*   STEP3 - SET UP PARAMETERS
//*
//*******************************************************************
//STEP3   EXEC ACBJBAOB,
//        TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//TEMPFILE  DD  DSN=&&TEMPFILE,DISP=(NEW,PASS),
//  SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ1 DISPLAY +
SCDS(TEMP.SCDS) +
MGMTCLAS() +
)
/*
```

Figure 153. Sample JCL for ACBJBAJ1 (Part 6 of 6)

Storage Group Volume Add/Delete: ACBQBAI9

SYS1.SACBCNTL member ACBJBAIB calls ACBQBAI9 to add SMS volumes from a storage group. When performing an ADD or a DELETE on an SMS volume in an SMS complex, run ACBQBAI9 for each MVS image in the SMS complex.

See Figure 154 on page 370 for the sample JCL and parameters.

Use one of the following DDNAMES to indicate which operation is to occur:

VOLADD lists the volumes to add to a storage group in an SCDS
VOLALT lists the volumes to be altered in a storage group in an SCDS
VOLDEL lists the volumes to delete from a storage group in an SCDS

Sample JCL for Storage Group Volume Add/Delete

To delete a volume, specify the following:

- SCDSNAME
- SG
- VOL
- VOLDEL for the DDNAME

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO ADD NEW VOLUMES AND THEIR STATUS                  *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*      CHANGE PARAMETERS                                          *
//*                                                                 *
//*      TEMPFILE - LISTING DATA SET (OUTPUT)                       *
//*      VOLADD   - VOLUMES TO BE ADDED (INPUT)                     *
//*                                                                 *
//*   PARAMETERS:                                                   *
//*                                                                 *
//* Required Fields:                                       2@WA49380 *
//*                                                                 *
//*      SCDSNAME - SOURCE CONTROL DATA SET                         *
//*      VOL      - VOLUME                                          *
//*      SG       - STORAGE GROUP                                   *
//*                                                             3@WA49380 *
//* Optional Fields:                                                *
//*                                                                 *
//*      STATUS   - STATUS (ENABLE/NOTCON/DISALL/DISNEW/QUIALL/     *
//*                         QUINEW)                                 *
//*                Up to 32 statuses can be specified separated by  *
//*               commas to match the 32 systems.  If a status is   *
//*               skipped, the system status that falls in between 2 *
//*               commas will have default value of ENABLE.  @WA29937*
//*                                                                 *
//*      UPDHLVLSCDS - When modifying an SCDS, that was formatted   *
//*                    with a higher level of SMS, using a lower    *
//*                    level of SMS will make this application fail *
//*                    unless you specify the UPDHLVLSCDS parameter *
//*                    as 'Y'.  Default is 'N'.             @WA41441*
//*                                                                 *
//*                      If specified, this should be the first     *
//*                    parameter on  either VOLDEL or ADDVOL DD     *
//*                    names.                               @WA41441*
//*                                                                 *
//*                    Possible values : Y/N/BLANK          @WA41441*
```

Figure 154. Sample JCL for ACBJBAIB (Part 1 of 2)

```
//*                                                                  *
//*   To DELETE volume, specify SCDSNAME, VOL and SG,                *
//*    and use VOLDEL for DDname.                            @WA29937*
//*                                                                  *
//*   To ALTER Volume, specify all the parameters as in VOLUME 11@U1A*
//*   ADD and use VOLALT as DDname.                                  *
//*                                                                  *
//*   Specify '+' as the continuation character.  When splitting a   *
//*    word or a parameter to the next line, and there should not    *
//*    be any spaces, don't leave any blank spaces before the        *
//*    continuation character '+'.                           @WA41026*
//*                                                                  *
//*   If you specify all the three types of commands DELETE, ADD and *
//*   DEFINE then DELETEs are processed first, ALTERs are processed  *
//*   next and ADDs are processed finally irrespective of the order  *
//*   of the commands in the JOB.                                    *
//*******************************************************************
//ADDVOL1 EXEC ACBJBAOB,
//        PLIB1='SYS1.DGTPLIB',
//        TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//TEMPFILE  DD  DSN=&&VOLADDS,DISP=(NEW,KEEP),
//  SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAI9) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//VOLADD  DD  *
UPDHLVLSCDS()
SCDSNAME(MYCDS) VOL(SMS900) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS901) SG(PRIMARY) STATUS(DISALL,+
ENABLE,,DISNEW,,NOTCON,DISALL,ENABLE,QUIALL)
SCDSNAME(MYCDS) VOL(SMS902) SG(PRIMARY) +
STATUS(DISNEW)
SCDSNAME(MYCDS) VOL(SMS903) SG(PRIMARY) +
STATUS(DISALL,DISNEW,ENABLE,NOTCON,DISALL,DISNEW,DISALL,QUIALL,+
DISALL,QUINEW,ENABLE,NOTCON,QUIALL,DISNEW,DISALL,QUIALL,+
DISNEW,ENABLE,ENABLE,DISNEW,QUIALL,DISALL,DISALL,NOTCON,+
NOTCON,QUINEW,ENABLE,NOTCON,QUIALL,DISNEW,ENABLE,QUIALL)
SCDSNAME(MYCDS) VOL(SMS904) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS905) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS906) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS907) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS908) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS909) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS910) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS911) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS912) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS913) SG(PRIMARY) STATUS(ENABLE)
/*
//VOLALT  DD  *                                            /* 4@U1A*/
/*
//VOLDEL  DD *
/*
//ADDVOL2 EXEC ACBJBAOB,
//        PLIB1='SYS1.DGTPLIB',
//        TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD DSN=&&VOLADDS,DISP=(OLD,DELETE)
//*******************************************************************
```

Figure 154. Sample JCL for ACBJBAIB (Part 2 of 2)

ACS Routine Translate: ACBQBAO1

You can use the ISMF translate option to translate the ACS routines and store them in an SCDS. After successful translation and validation, that SCDS can be activated and used by the system to manage storage. ACBQBAO1 is called by the following SYS1.SACBCNTL members to translate the ACS routines:

ACBJBAC2 Translate ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility

ACBJBAOQ Translate ACS routines into an SCDS, followed by validating the SCDS

See Figure 155 for the sample JCL and parameters.

Sample JCL for ACS Routine Translate

```
//*******************************************************************
//*                                                                 *
//*  SAMPLE JCL TO DO FOUR OPEARTIONS:                              *
//*                                                                 *
//*     1. TRANSLATE ACS ROUTINES (ISMF OPTION 7.2)                 *
//*     2. VALIDATE ACS ROUTINES (ISMF OPTION 7.3)                  *
//*     3. TEST ACS ROUTINES (ISMF OPTION 7.4.3)                    *
//*     4. COMPARE BASE & NEW ACS LISTINGS                          *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*     CHANGE JOBCARD                                              *
//*     CHANGE PREFIX                                               *
//*     CHANGE PARAMETERS                                           *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* TRANSLATE STEP:                                                 *
//*                                                                 *
//*  ACSSRC      - PDS CONTAINING ACS ROUTINES TO BE                *
//*                TRANSLATED (INPUT)                               *
//*  MEMBER      - MEMBER NAME OF THE ROUTINE TO BE                 *
//*                TRANSLATED (INPUT)                               *
//*  SCDSNAME    - NAME OF SCDS INTO WHICH THE ACS ROUTINES ARE     *
//*                TO BE TRANSLATED (OUTPUT)                        *
//*  LISTNAME    - TRANSLATE LISTING (OUTPUT)                       *
//*  UPDHLVLSCDS - CONFIRM OPERATION ON AN UPLEVEL SCDS      @WA41441*
//*                When modifying an SCDS, that was formatted with a *
//*                higher level of SMS, using a lower level of SMS   *
//*                will make this application fail unless you        *
//*                specify the UPDHLVLSCDS parameter as 'Y'.         *
//*                Default is 'N'.                          @WA41441*
//*                Possible values : Y/N/BLANK              @WA41441*
//*******************************************************************
//TRANSLAT EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
```

Figure 155. Sample JCL for ACBJBAC2 (Part 1 of 4)

```
DEL DATACLAS.LISTING
DEL MGMTCLAS.LISTING
DEL STORGRP.LISTING
DEL STORCLAS.LISTING
ISPSTART CMD(ACBQBAO1 +
ACSSRC(ACS.SOURCE) MEMBER(DATACLAS) +
SCDSNAME(MYSCDS) LISTNAME(DATACLAS.LISTING) +
UPDHLVLSCDS()) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

ISPSTART CMD(ACBQBAO1 +
ACSSRC(ACS.SOURCE) MEMBER(STORCLAS) +
SCDSNAME(MYSCDS) LISTNAME(STORCLAS.LISTING) +
UPDHLVLSCDS()) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

ISPSTART CMD(ACBQBAO1 +
ACSSRC(ACS.SOURCE) MEMBER(MGMTCLAS) +
SCDSNAME(MYSCDS) LISTNAME(MGMTCLAS.LISTING) +
UPDHLVLSCDS()) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

ISPSTART CMD(ACBQBAO1 +
ACSSRC(ACS.SOURCE) MEMBER(STORGRP) +
SCDSNAME(MYSCDS) LISTNAME(STORGRP.LISTING) +
UPDHLVLSCDS()) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)

/*
//*******************************************************************
//*                                                                 *
//*   COPY TRANSLATE LISTINGS                                       *
//*                                                                 *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          *
//*    SYSUT2 - OUTPUT                                              *
//*                                                                 *
//*******************************************************************
//TRANGEN  EXEC  PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.DATACLAS.LISTING,DISP=SHR
//         DD  DSN=IBMUSER.STORCLAS.LISTING,DISP=SHR
//         DD  DSN=IBMUSER.MGMTCLAS.LISTING,DISP=SHR
//         DD  DSN=IBMUSER.STORGRP.LISTING,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*******************************************************************
//*                                                                 *
//*   VALIDATE STEP:                                                *
//*                                                                 *
//*  SCDSNAME    - NAME OF SCDS THAT CONTAINS THE TRANSLATED ACS    *
//*                ROUTINES TO BE VALIDATED (INPUT)                 *
//*  TYPE        - TYPE OF ACS ROUTINE TO BE VALIDATED (INPUT)      *
//*  LISTNAME    - VALIDATE LISTING (OUTPUT)                        *
//*  UPDHLVLSCDS - CONFIRM OPERATION ON AN UPLEVEL SCDS(Y/N) @WA41441*
//*                                                                 *
//*******************************************************************
//VALIDAT  EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
```

Figure 155. Sample JCL for ACBJBAC2 (Part 2 of 4)

```
DEL VALIDAT.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAO2 SCDSNAME(MYSCDS) TYPE(*) +
LISTNAME(VALIDAT.LISTING) +
UPDHLVLSCDS()) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
//*                                                                 *
//*   COPY VALIDATE LISTING                                         *
//*                                                                 *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          *
//*    SYSUT2 - OUTPUT                                              *
//*                                                                 *
//*******************************************************************
//VALGEN   EXEC  PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.VALIDAT.LISTING,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*******************************************************************
//*                                                                 *
//*   TEST STEP                                                     *
//*                                                                 *
//*    SCDSNAME - NAME OF SCDS (INPUT)                              *
//*    TESTBED  - PDS CONTAINING TEST CASES (INPUT)                 *
//*    MEMBER   - MEMBERS TO BE TESTED IN TESTBED (INPUT)           *
//*    DC,SC,MC,SG - ROUTINES TO BE TESTED (INPUT)                  *
//*    LISTNAME - TEST LISTING (OUTPUT)                             *
//*                                                                 *
//*******************************************************************
//TEST     EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
DEL NEW.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAIA +
SCDSNAME(MYSCDS) +
TESTBED(TESTCASE.LIBRARY) MEMBER(*) +
DC(Y) SC(Y) MC(Y) SG(Y) +
LISTNAME(NEW.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
//*                                                                 *
//*   COPY TEST LISTING                                             *
//*                                                                 *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          *
//*    SYSUT2 - OUTPUT                                              *
//*                                                                 *
//*******************************************************************
//TSTGEN   EXEC  PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.NEW.LISTING,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
```

Figure 155. Sample JCL for ACBJBAC2 (Part 3 of 4)

```
//*******************************************************************
//*                                                                 *
//* COMPARE ACS LISTINGS                                            *
//*                                                                 *
//*   PARAMETERS:                                                   *
//*                                                                 *
//*     BASELIST - BASE ACS TEST LIST (INPUT)                       *
//*     NEWLIST  - NEW ACS TEST LIST (INPUT)                        *
//*     TESTBED  - PDS CONTAINING TEST CASES (REFERENCE INPUT)      *
//*     RSLTDSN  - COMPARISON RESULTS DATA SET (OUTPUT)             *
//*     XCPTPDS  - EXCEPTION TEST CASE PDS (OUTPUT)                 *
//*     XCPSPACE - SPACE values of Except DS (Optional)    3a@WA32832*
//*       Values: (Primary Tracks,Secondary Tracks,Directory Blocks) *
//*         which are positional and optional. Defaults: (3,1,20).  *
//*                                                                 *
//*   NOTE:  If you receive message IEC217I B14-0C on your exception *
//*       data set, you need to increase your data set size by using *
//*       the XCSPACE parameter (specially the directory blocks)    *
//*                                                                 *
//*******************************************************************
//COMPARE  EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
DEL PDS4
DEL RESULTS.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAC1 +
BASELIST(BASE.TESTLIST) +
NEWLIST(NEW.TESTLIST) +
TESTBED(TESTCASE.LIBRARY) +
RSLTDSN(COMPARE.LISTING) +
XCPTPDS(TESTCASE.EXCP) +
XCPSPACE(5,3,30)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
//*                                                                 *
//*   COPY COMPARISON RESULTS                                       *
//*                                                                 *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          *
//*    SYSUT2 - OUTPUT                                              *
//*                                                                 *
//*******************************************************************
//CMPGEN   EXEC PGM=IEBGENER
//SYSUT1   DD DISP=SHR,DSN=IBMUSER.COMPARE.LISTING
//SYSUT2   DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
//*
```

Figure 155. Sample JCL for ACBJBAC2 (Part 4 of 4)

SCDS Validation: ACBQBAO2

ACBQBAO2 is called by the following SYS1.SACBCNTL members to validate the constructs:

ACBJBAOQ Translate all 4 ACS routines into an SCDS and validate SCDS

ACBJBAC2 Translate all 4 ACS routines into an SCDS, validate the SCDS, run option 7.4.3 (test ACS routines) in batch, run the NaviQuest ACS comparison utility

ACBJBAOS ISMF batch EXEC for validating the SCDS

See Figure 156 on page 377 for the sample JCL and parameters.

Sample JCL for SCDS Validation

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO PERFORM ISMF ACS VALIDATE IN BATCH                *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*      CHANGE PARAMETERS                                          *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//*   VALIDATE STEP:                                                *
//*                                                                 *
//*    SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED ACS     *
//*               ROUTINES TO BE VALIDATED (INPUT)                  *
//*    TYPE     - TYPE OF ACS ROUTINE TO BE VALIDATED (INPUT)       *
//*    LISTNAME - VALIDATE LISTING (OUTPUT)                         *
//*    UPDHLVLSCDS - When modifying an SCDS, that was formatted with *
//*                  a higher level of SMS, using a lower level of  *
//*                  SMS will make this application fail unless you *
//*                  specify the UPDHLVLSCDS parameter as 'Y'.      *
//*                  Default is 'N'.                        @WA41441*
//*                                                                 *
//*                  Possible values : Y/N/BLANK            @WA41441*
//*******************************************************************
//VALIDAT  EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
DEL VALIDAT.LISTING
ISPSTART CMD(ACBQBAO2 +
SCDSNAME(DSNREPN) TYPE(*) +
LISTNAME(VALIDAT.LISTING) +
UPDHLVLSCDS()) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
//*                                                                 *
//* COPY THE VALIDATE LISTING DATA SET                              *
//*                                                                 *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          *
//*    SYSUT2 - OUTPUT                                              *
//*                                                                 *
//*******************************************************************
//STEP2  EXEC  PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.VALIDAT.LISTING,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*******************************************************************
```

Figure 156. Sample JCL for ACBJBAOS

Test ACS Routines: ACBQBAIA

ACBQBAIA is called by the following SYS1.SACBCNTL members to test the ACS routines in batch:

ACBJBAC2 Translate all 4 ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility

ACBJBAIC Test ACS routines

See Figure 157 on page 379 for the sample JCL and parameters.

Sample JCL for Test ACS Routines

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO TEST ACS ROUTINES IN BATCH                        *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//*   TEST STEP                                                     *
//*                                                                 *
//*    SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED,        *
//*               VALIDATED ACS ROUTINES TO BE TESTED (INPUT)       *
//*    TESTBED  - PDS CONTAINING TEST CASES THAT THE ACS ROUTINES   *
//*               SHOULD BE TESTED FOR (INPUT)                      *
//*    MEMBER   - MEMBERS TO BE TESTED IN TESTBED (INPUT)           *
//*    DC,SC,MC,SG - ROUTINES TO BE TESTED Y OR N (INPUT)           *
//*    LISTNAME - TEST LISTING (OUTPUT)                             *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//TESTACS  EXEC ACBJBAOB,
//         PLIB1='SYS1.DGTPLIB',
//         TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
DEL NEW.TESTLIST
ISPSTART CMD(ACBQBAIA +
SCDSNAME(MYSCDS) +
TESTBED(TESTCASE.LIBRARY) MEMBER(*) +
LISTNAME(NEW.TESTLIST) +
DC(Y) SC(Y) MC(Y) SG(Y)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************/
//*                                                                 */
//*   COPY TEST LISTING                                             */
//*                                                                 */
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          */
//*    SYSUT2 - OUTPUT                                              */
//*                                                                 */
//*******************************************************************/
//TESTGEN  EXEC  PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.NEW.TESTLIST,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
```

Figure 157. Sample JCL for ACBJBAIC

Generate Test Cases from ISMF-saved Data Set Lists: ACBQBAG3

ACBQBAG3 is called by SYS1.SACBCNTL member ACBJBAG2 to take an ISMF-saved data set table and generate test case members in a PDS library based on the attributes of the data sets in the ISMF table.

See Figure 158 for the sample JCL and parameters.

Recommendations and Restrictions:

- Saving tables of temporary data sets might produce errors in the bulk test case generate option (11.1.1). Instead, generate test cases from the ACSTST program for temporary data sets.
- Before generating the list, set the ACQUIRE DATA FROM VOLUME and ACQUIRE DATA IF DFHSM options under the ISMF data set selection entry panel to Y.
- The MULTVOL variable is always set to YES in an ISMF table if the data set has not been opened at the time the table is saved. The value is set correctly at OPEN time. This can sometime cause errors in the bulk test case generator.

Sample JCL for Generating Test Cases from ISMF-saved Data Set Lists

```
//*********************************************************************
//*                                                                   *
//* GENERATE TEST CASES FROM A PREVIOUSLY SAVED ISMF DATA SET LIST    *
//*                                                                   *
//*     TABLENM  - SAVED ISMF DATA SET LIST (INPUT)                   *
//*     PRFX     - MEMBER NAME PREFIX                                 *
//*     TESTPDS  - TEST CASE PDS (OUTPUT)                             *
//*     REPLACE  - REPLACE CONTENTS IF DSN EXISTS                     *
//*     INTEST   - DEBUG MODE YES/NO (KEEP IT AS NO)                  *
//*                                                                   *
//*********************************************************************
//GENTC   EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAG3 TABLENM(DSNLIST) +
INTEST(NO) PRFX(TEST) +
TESTPDS('IBMUSER.TESTCASE.LIBRARY') REPLACE(Y)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*********************************************************************
```

Figure 158. Sample JCL for ACBJBAG2

Generate Test Cases from DCOLLECT Data: ACBQBAG1

ACBQBAG1 is called by SYS1.SACBCNTL member ACBJBAG1 to generate test cases from DCOLLECT data.

See Figure 159 for the sample JCL and parameters:

Sample JCL for Generating Test Cases from DCOLLECT Data

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE TEST CASES FROM DCOLLECT DATA            *
//*                                                                 *
//* INSTRUCTIONS BEFORE SUBMITTING:                                 *
//*                                                                 *
//*   CHANGE JOBCARD                                                *
//*   CHANGE PREFIX                                                 *
//*   CHANGE PARAMETERS                                             *
//*                                                                 *
//* THE PARAMETERS TO ACBQBAG1 ARE AS FOLLOWS:                      *
//*                                                                 *
//*   PARAMETER 1 IS DATA SET CONTAINING DCOLLECT DATA (INPUT)      *
//*   PARAMETER 2 IS NUMBER OF TEST CASES TO BE GENERATED           *
//*   PARAMETER 3 IS MEMBER NAME PREFIX                             *
//*   PARAMETER 4 IS TEST CASE PDS (OUTPUT)                         *
//*   PARAMETER 5 IS REPLACE CONTENTS IF DSN EXISTS                 *
//*                                                                 *
//*******************************************************************
//GENTEST EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAG1 'IBMUSER.DCOLLECT.DATA' 10 TEST +
'IBMUSER.TESTCASE.LIBRARY' Y) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
```

Figure 159. Sample JCL for ACBJBAG1

Generate Test Cases from SMF Data

SYS1.SACBCNTL member ACBJBAI1 contains JCL to generate test cases from SMF type 127 type records in batch.

You will need to provide the names of the input and output data sets on the INDD and OUTDD DD statements:

INDD Data set containing SMF data (input)

OUTDD Test case PDS (output)

See Figure 160 for the sample JCL and parameters.

SMF type 127 records are written by storage class exit IGDACSSC. Use a utility like IFASMFDP to unload the SMF type 127 records from the SMF log data sets. You can also use ACSTST program (from CBIPO).

Sample JCL for Generating Test Cases from SMF Data

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE TEST CASES FROM SMF TYPE 127 RECORDS     *
//* WRITTEN BY THE STORAGE CLASS EXIT IGDACSSC; USE A STANDARD      *
//* UTILITY LIKE IFASMFDP TO UNLOAD THE TYPE 127 RECORDS FROM THE   *
//* SMF LOG DATA SETS; ALSO ACSTST PROGRAM (FROM CBIPO) SHOULD BE   *
//* AVAILABLE                                                       *
//*                                                                 *
//*    INDD  - DATA SET CONTAINING SMF DATA (INPUT)                 *
//*    OUTDD - TEST CASE PDS (OUTPUT)                               *
//*                                                                 *
//*******************************************************************
//GENTEST EXEC PGM=ACSTST,REGION=512K,
//             COND=(0,NE,DATA1)
//SYSPRINT DD  SYSOUT=*
//INDD     DD  DSN=IBMUSER.SMF.TYPE127.DATA,DISP=SHR
//OUTDD    DD  DSN=IBMUSER.TESTCASE.LIBRARY,DISP=(NEW,CATLG,DELETE),
//             SPACE=(80,(1250,200,50),,,ROUND),
//             UNIT=3380,VOL=SER=M4RS05,
//             DCB=(BLKSIZE=80,LRECL=80,RECFM=F)
//*******************************************************************
```

Figure 160. Sample JCL for ACBJBAI1

Generate Test Cases from VMA Extract Data: ACBQBAO3

ACBQBAO3 is called by SYS1.SACBCNTL member ACBJBAOW to generate test cases from records in the VMA extract file of previously created VMA data.

See Figure 161 for the sample JCL and parameters.

Tip: You can also use the ACSTST program to generate test cases.

Sample JCL for Generating Test Cases from VMA

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE TEST CASES FROM VMA EXTRACT DATA         *
//*                                                                 *
//* INSTRUCTIONS BEFORE SUBMITTING:                                 *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*      CHANGE PARAMETERS                                          *
//*                                                                 *
//* PARAMETERS:                                                     *
//*                                                                 *
//*   PARAMETER 1 - DATA SET CONTAINING VMA EXTRACT DATA            *
//*   PARAMETER 2 - NUMBER OF TEST CASES TO BE GENERATED            *
//*   PARAMETER 3 - MEMBER NAME PREFIX                              *
//*   PARAMETER 4 - PROGRAM NAME TO FILTER ON                       *
//*                                                                 *
//*******************************************************************
//TESTGEN EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAO3 'IBMUSER.VMA.DATA' 100 TEST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
```

Figure 161. Sample JCL for ACBJBAOW

ACS Test Listings Comparison: ACBQBAC1

ACBQBAC1 is called by the following SYS1.SACBCNTL members to perform a detailed comparison of the differences between the ″base″ and ″new″ ACS listing:

ACBJBAC1 Run the NaviQuest ACS Comparison utility.

ACBJBAC2 Translate all 4 ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility.

See Figure 162 for the sample JCL and parameters:

Sample JCL for Test Listings Comparison

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO COMPARE ACS TEST LISTINGS IN BATCH                *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*     CHANGE JOBCARD                                              *
//*     CHANGE PREFIX                                               *
//*     CHANGE PARAMETERS                                           *
//*                                                                 *
//*   PARAMETERS:                                                   *
//*                                                                 *
//*     BASELIST - BASE ACS TEST LISTING (INPUT)                    *
//*     NEWLIST  - NEW ACS TEST LISTING (INPUT)                     *
//*     TESTBED  - TEST CASE PDS (REFERENCE INPUT)                  *
//*     RSLTDSN  - COMPARISON RESULTS DATA SET (OUTPUT)             *
//*     XCPTPDS  - EXCEPTION TEST CASE PDS (OUTPUT)                 *
//*     XCPSPACE - SPACE values of Except DS (Optional)    3a@WA32832*
//*       Values: (Primary Tracks,Secondary Tracks,Directory Blocks) *
//*         which are positional and optional. Defaults: (3,1,20).  *
//*                                                                 *
//*   NOTE:  If you receive message IEC217I B14-0C on your exception *
//*       data set, you need to increase your data set size by using *
//*       the XCSPACE parameter (specially the directory blocks)    *
//*                                                                 *
//*******************************************************************
//CMPRSTEP EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
 PROFILE PREFIX(IBMUSER)
DEL COMPARE.LISTING
DEL TESTCASE.EXCP
```

Figure 162. Sample JCL for ACBJBAC1 (Part 1 of 2)

```
ISPSTART CMD(%ACBQBAC1 +
BASELIST(BASE.TESTLIST) +
NEWLIST(NEW.TESTLIST) +
TESTBED(TESTCASE.LIBRARY) +
RSLTDSN(COMPARE.LISTING) +
XCPTPDS(TESTCASE.EXCP) +
XCPSPACE(5,3,30)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*******************************************************************
//*                                                                 *
//* COPY THE COMPARISON LISTINGS DATA SET                           *
//*                                                                 *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          *
//*    SYSUT2 - OUTPUT                                              *
//*                                                                 *
//*******************************************************************
//REPGEN   EXEC PGM=IEBGENER
//SYSUT1   DD DISP=SHR,DSN=IBMUSER.COMPARE.LISTING
//SYSUT2   DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
//
```

Figure 162. Sample JCL for ACBJBAC1 (Part 2 of 2)

Enhanced ACS Test Listing: ACBQBAX1

ACBQBAX1 is called by SYS1.SACBCNTL member ACBJBAX1 to generate a detailed cross-reference listing report from the original ACS listing. The cross-reference listing report is used to help determine where there are logic errors in an ACS routine. The report can include the data set name, unit type, data set size, expiration date, job name, and program name for each exception test case.

See Figure 163 for the sample JCL and parameters.

Use the following parameters with the ACBQBAX1:

Parameters	Description
DSNINFO(YIN)	Includes the data set name in the cross-reference
UNITINFO(YIN)	Includes the unit name in the cross-reference
SIZEINFO(YIN)	Includes size information (in K) in the cross-reference
EXPTINFO(YIN)	Includes expiration date information in the cross-reference
JOBINFO(YIN)	Includes jobname information in the cross-reference
PGMINFO(YIN)	Includes program name information in the cross-reference

Sample JCL for Generating an Enhanced ACS Test Listing

```
//* SAMPLE JCL TO GENERATE ENHANCED ACS TEST LISTING IN BATCH        *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* STEP TO GENERATE ENHANCED ACS TEST LISTING                      *
//*                                                                 *
//* PARAMETERS:                                                     *
//*                                                                 *
//*      ACSDSN   - ACS TEST LISTING (INPUT)                        *
//*      XREFDSN  - ENHANCED ACS TEST LISTING (OUTPUT)              *
//*      XREFSPC  - The Space parameters can be specified  12@WA42064*
//*                 for XREFDSN dataset using this parameter.       *
//*                                                                 *
//*                 Syntax ==> XREFSPC(PRI,SEC) Where               *
//*                            PRI -> Primary space in Tracks       *
//*                            SEC -> Secondary space in Tracks     *
//*                                   to be allocated.              *
//*                 Eg. XREFSPC(10,20)                              *
//*                                                                 *
//*                 If this parameter is not specified, by default  *
//*                 ACSDSN dataset space parameters are used to     *
//*                 allocate XREFDSN dataset.                       *
//*                                                                 *
//*      OUTPUT SELECTION CRITERIA: (DATA COMES FROM TEST CASE PDS) *
//*         DSNINFO  - DSN   - DATA SET NAME                        *
//*         UNITINFO - UNIT  - UNIT ON WHICH THE DATA SET RESIDES   *
//*         SIZEINFO - SIZE  - SIZE OF DATA SET                     *
//*         EXPTINFO - EXPDT - EXPIRY DATE OF THE DATA SET          *
//*         JOBINFO  - JOB   - JOB WHICH ALLOCATED THE DATA SET     *
//*         PGMINFO  - PGM   - PROGRAM WHICH ALLOCATED THE DATA SET *
//*                                                                 *
//*******************************************************************
```

Figure 163. Sample JCL for ACBJBAX1 (Part 1 of 2)

```
//GENREP  EXEC  ACBJBAOB,
//        PLIB1=SYS1.DGTPLIB,
//        TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
DEL RSLT1
%ACBQBAX1 +
ACSDSN(BASE.LISTING) +
XREFDSN(RSLT1) +
XREFSPC() +
DSNINFO(Y) +
UNITINFO(Y) +
SIZEINFO(Y) +
EXPTINFO(Y) +
JOBINFO(Y) +
PGMINFO(Y)
/*
//*******************************************************************
//*                                                                 *
//* COPY ENHANCED ACS TEST LISTING                                  *
//*                                                                 *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                          *
//*    SYSUT2 - OUTPUT                                              *
//*                                                                 *
//*******************************************************************
//REPGEN   EXEC PGM=IEBGENER
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=IBMUSER.ENHANCED.LISTING
//SYSUT2   DD SYSOUT=*
//*
```

Figure 163. Sample JCL for ACBJBAX1 (Part 2 of 2)

Update Test Cases with Expected Results: ACBQBAU1

ACBQBAU1 is called by SYS1.SACBCNTL member ACBJBAU2 to update the testbed library with the expected results for the new data set type that has been placed in the *new* ACS listing.

Tip: ACBQBAU1 is I/O intensive, as it must update each test case member in the PDS with the new expected results.

See Figure 164 for the sample JCL and parameters:

Sample JCL to Update Test Cases

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO UPDATE TEST CASES WITH TEST RESULTS IN BATCH      *
//*                                                                 *
//*    INSTRUCTIONS BEFORE SUBMITTING:                              *
//*                                                                 *
//*       CHANGE JOBCARD                                            *
//*       CHANGE PREFIX                                             *
//*       CHANGE PARAMETERS                                         *
//*                                                                 *
//*    PARAMETERS:                                                  *
//*                                                                 *
//*       NEWLIST  - NEW ACS TEST LISTING                           *
//*       TESTBED  - TEST CASE PDS                                  *
//*       XCPTDSN  - EXCEPTION TEST CASE PDS                        *
//*       RSLTDSN  - COMPARISON RESULTS DATA SET                    *
//*                                                                 *
//*******************************************************************
//UPDTE   EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAU1 +
NEWLIST(NEW.TESTLIST) +
XCPTPDS(TESTCASE.EXCP) +
TESTBED(TESTCASE.LIBRARY) +
RSLTDSN(COMPARE.LISTING)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
```

Figure 164. Sample JCL for ACBJBAU2

Generate Report from ISMF-saved Data Set List: ACBQBAR1

ACBQBAR1 EXEC is called by SYS1.SACBCNTL member ACBJBAOD to generate a flat file from an ISMF-saved data set table, listing the fields of your choice, in the order you specify.

See Figure 165 on page 391 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR1 EXEC. Each parameter must be on a separate line.

Parameters	Description
%NOTUSED	Prints the amount of space *not* used for the data set.
ALLOCSP|ALLOCSPC	Prints the data set's allocated space.
ALLOCUSED	Prints the amount of used space for the data set.
BACKUP|LASTBKUP	Prints the date of the last backup for the data set.
BLKSIZE|CISIZE	Prints the BLKSIZE or CISIZE of the data set.
BLKUNUSED	Prints the number of unused blocks for the data set.
CCSIDDES	CCSID description
CFMONST	CF monitor status
CHANGE|CHGIND	Is the change indicator set for the data set?
CFSTATUS	CF status indicator
COMPRESS	Compressed format
CREATE|CREATEDT	Prints the data set's create date.
CSETNAME	CF cache set name
CSTRNAME	CF cache structure name
DATACLAS|DC	Prints the data class for the data set, if there is one.
DDMATTR	DDM attribute
DEVTYPE|DEVICE	Prints the device type that the data set resides on.
DSNAMETY	Dataset name type
DSNAME|DSN	Prints the data set name.
DSNLENGTH=nn	Limits print of data set name to nn characters. Defaults to 44 characters.
DSORG	Prints the data set's organization.
ENTRYTYPE|ENTRY	Prints the entry type of data set.
ENVIRONMENT|ENVIR	Prints the data set's environment, managed or unmanaged.
EXPIRE|EXPIREDT	Prints the data set's expiration date.
EXTNUM	Prints the data set's extension number.
LASTREF|LASTREFDT	Prints the last reference date of the data set.
LRECL	Prints the LRECL of the data set.
MGMTCLAS|MC	Prints the management class for the data set, if there is one.
MULTVOL	Is the data set a part of a multivolume data set?
NUMEXT|EXTNUM	Prints the number of extents for the data Set.
NUMSTR	Number of stripes
OPTIMAL	Prints the optimal block size of the data Set.

Parameters	Description
OWNER	Prints the owner of the data set, if there is one.
PAGELENGTH=nn	Sets page length for the report (default is 60).
PDSE	Is the data set a PDSE?
REBLOCK|REBLK	Is the data set reblockable?
RECFM|RECFMT	Prints the record format of the data set.
SECALLOC|ALLOCSEC	Prints the secondary allocation amount for the data set.
STORCLAS|SC	Prints the storage class for the data set, if there is one.
TITLE=xxx	Prints a title for the report. Title does *not* need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
TOTALS	Prints totals for the data set space allocations. If this parameter is not specified, totals does not print.
UNIT|ALLOCUNIT	Prints the allocation unit (TRKS, CYLS, and for forth) for the data set.
USERRED%	% user data reduction
VOLSER	Prints the data set's volume serial number.

Sample JCL for Generating a Report from ISMF-saved Data Set List

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE DATA SET REPORT FROM A PREVIOUSLY SAVED  *
//* ISMF DATA SET LIST                                              *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*      CHANGE PARAMETERS                                          *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* DELETE STEP TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY  *
//*                                                                 *
//*******************************************************************
//DELREP   EXEC  PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
 DELETE IBMUSER.DATASET.REPORT
/*
//*******************************************************************
//*                                                                 *
//* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED   *
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY    *
//*       DATA SET.                                                 *
//*                                                                 *
//*******************************************************************
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD  DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
//    BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN  DD *
/*
//*******************************************************************
//*                                                                 *
//* DATA SET REPORT GENERATION STEP                                 *
//*                                                                 *
//*    PARAMETER FOLLOWING ACBQBAR1 - SAVED ISMF LIST (INPUT)       *
//*    ISPFILE - DATA SET REPORT (OUTPUT, FROM ALCISPFL STEP)       *
//*    SYSIN   - KEY WORDS SPECIFYING DATA IN THE REPORT            *
//*                                                                 *
//*    The Following parameters can be specified in SYSIN      42@U1A*
//*                                                                 *
//*    %NOTUSED                ->   % Space Not Used.               *
//*    ALLOCSP    | ALLOCSPC   ->   Allocated Space.                *
//*    ALLOCUSED               ->   Used Space.                     *
//*    BACKUP     | LASTBKUP   ->   Last Backup Date.               *
//*    BLKSIZE    | CISIZE     ->   Block/CI Size.                  *
//*    BLKUNUSED               ->   Blocks Unused.                  *
//*    CCSIDDES                ->   CCSID Description.              *
//*    CFMONST                 ->   CF Monitor Status.              *
//*    CFSTATUS                ->   CF Status Indicator.            *
//*    CHANGE     | CHGIND     ->   Change Indicator.               *
//*    COMPRESS                ->   Compressed Format.              *
//*    CREATE     | CREATEDT   ->   Creation Date.                  *
//*    CSETNAME                ->   CF Cache set name.          1@U2C*
//*    CSTRNAME                ->   CF Cache Structure Name.        *
```

Figure 165. Sample JCL for ACBJBAOD (Part 1 of 2)

```
//*    DC         | DATACLAS   ->   Data Class Name.                 *
//*    DDMATTR                 ->   DDM attribute.                   *
//*    DEVICE     | DEVTYPE    ->   Device Type.                     *
//*    DSNAME     | DSN        ->   Dataset Name.                    *
//*    DSNAMETY                ->   Dataset Name Type.               *
//*    DSORG                   ->   Data Set Organization.           *
//*    ENTRY      | ENTRYTYPE  ->   Dataset Entry Type.              *
//*    ENVIRONMANT| ENVIR      ->   Dataset Environment.             *
//*    EXPIRE     | EXPIREDT   ->   Expiration Date.                 *
//*    LASTREF    | LASTREFDT  ->   Last referenced Date.            *
//*    LRECL                   ->   Record Length.                   *
//*    MC         | MGMTCLAS   ->   Management Class Name.           *
//*    MULTVOL                 ->   Multi Volume Status.             *
//*    NUMEXT     | EXTNUM     ->   Number of Extents.               *
//*    NUMSTR                  ->   Number of Stripes.               *
//*    OPTIMAL                 ->   Optimal Block/CI Size.           *
//*    OWNER                   ->   Owner.                           *
//*    REBLK      | REBLOCK    ->   Reblock Indicator.               *
//*    RECFM      | RECFMT     ->   Record Format.                   *
//*    SC         | STORCLAS   ->   Storage Class Name.              *
//*    SECALLOC   | ALLOCSEC   ->   Secondary Allocation.            *
//*    TOTALS                  ->                                    *
//*    UNIT       | ALLOCUNI   ->   Allocation Unit.                 *
//*    USERRED%                ->   % User Data Reduction.           *
//*    VOLSER                  ->   Volume Serial.                   *
//*********************************************************************
//GENREP  EXEC ACBJBAOB,
//        PLIB1=SYS1.DGTPLIB,
//        TABL2=userid.TEST.ISPTABL
//ISPFILE DD  DSN=IBMUSER.DATASET.REPORT,DISP=OLD
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR1 DSNLIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD  *
TITLE=LIST OF DATASETS FOR HIGHLEVEL QUALIFIER IBMUSER
DSN
VOLSER
LASTREF
EXTNUM
TOTALS
/*
//*********************************************************************
//*                                                                   *
//* COPY THE DATA SET REPORT                                          *
//*                                                                   *
//*    SYSUT1 - INPUT (FROM PREVIOUS STEP)                            *
//*    SYSUT2 - OUTPUT                                                *
//*                                                                   *
//*********************************************************************
//STEP2  EXEC PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.DATASET.REPORT,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
```

Figure 165. Sample JCL for ACBJBAOD (Part 2 of 2)

Generate Report from ISMF-saved DASD Volume List: ACBQVAR1

ACBQVAR1 is called by SYS1.SACBCNTL member ACBJBAOF to generate a flat file from an ISMF-saved DASD volume table and lists the fields of your choice, in the order you specify.

See Figure 166 on page 394 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQVAR1 EXEC in batch. Each parameter must be on a separate line.

Parameters	Description
%FREE	Prints the % of free space on the volume.
ALLOCSPC | ALLOCSP	Prints the amount of allocated space on the volume.
CACHEFW	Does the volume have Cache Fast Write enabled for it?
CFVOLST	CF Volume Status
DASDFW	Does the volume have DASD FAST WRITE enabled for it?
DEVICE | DEVICETYPE	Prints the device type of the volume.
DEVNUM | ADDRESS	Prints the device number of the volume.
DUPLEX | DUPLEXST	Is the volume part of a dual-copy pair?
FRAG | FRAGINDX	Prints the fragmentation index of the Volume.
FREEDSCB | DSCBFREE	Prints the number of free data set control blocks in the VTOC of the volume.
FREESPC | FREESP	Prints the volume's free space in KB.
FREEVIRS | VIRSFREE	Prints the number of free VIRs in the indexed VTOC on the volume.
FREEXT | EXTFREE	Prints the number of free extents on the Vvolume.
INDXSTAT | STATINDX	Prints the status of the indexed VTOC on the volume.
LRGEXT | EXTLRG	Prints the size of the largest extent available on the volume.
OTHER | OTHERDEV	If duplex, what is the secondary volume?
PAGELENGTH=nn	Sets page length for the report (default is 60).
PHYSTAT | STATPHYS	Prints the physical status of the volume.
RDCACHE | RDSTAT	Does the volume have READ CACHE enabled for it?
SHARE | SHRDASD	Is the DASD volume genned as SHARED?
STORGRP | SG	Prints the storage group that the volume belongs to.
SUBSYS | SUBSYSID	Prints the controller's subsystem identifier.
TITLE=xxx	Prints a title for the report. Title does *not* need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
TOTALS	Prints totals for the volume. If this parameter is not specified, totals does not print.
USE | USEATTR	Prints the volumes use attribute (public and so on).
VOLSER	Prints the volume serial number.

Sample JCL for Generating a Report from ISMF-saved DASD Volume List

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE DASD VOLUME REPORT FROM A PREVIOUSLY     *
//* SAVED ISMF DASD VOLUME LIST                                     *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*       CHANGE JOBCARD                                            *
//*       CHANGE PREFIX                                             *
//*       CHANGE PARAMETERS                                         *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* DELETE STEP TO DELETE THE REPORT IF IT EXISTS ALREADY           *
//*                                                                 *
//*******************************************************************
//DELREP  EXEC  PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
 DELETE IBMUSER.DASDVOL.REPORT
/*
//*******************************************************************
//*                                                                 *
//* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED   *
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY    *
//*       DATA SET.                                                 *
//*                                                                 *
//*******************************************************************
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD  DSN=IBMUSER.DASDVOL.REPORT,DISP=(NEW,CATLG),
//    BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN  DD *
/*
//*******************************************************************
//*                                                                 *
//* DASD VOLUME REPORT GENERATION STEP                              *
//*                                                                 *
//*    PARAMETER FOLLOWING ACBQVAR1 - ISMF SAVED LIST (INPUT)       *
//*    ISPFILE - DASD VOLUME REPORT (OUTPUT, FROM ALCISPFL STEP)    *
//*    SYSIN   - KEY WORDS TO SPECIFY THE DATA IN THE REPORT        *
//*                                                                 *
//*    The following parameters can be specified in SYSIN.    25@U1A *
//*                                                                 *
//*    VOLSER                 ->    Volume Serial                   *
//*    FREESPC   | FREESP     ->    Free Space                      *
//*    %FREE                  ->    % Free                          *
//*    ALLOCSPC  | ALLOCSP    ->    Allocated Space                 *
//*    FRAG      | FRAGINDX   ->    Fragmentation Index             *
//*    LRGEXT    | EXTLRG     ->    Largest Extent                  *
//*    FREEXT    | EXTFREE    ->    Free Extents                    *
//*    INDXSTAT  | STATINDX   ->    Index Status                    *
//*    FREEDSCB  | DSCBFREE   ->    Free DSCBS                      *
//*    FREEVIRS  | VIRSFREE   ->    Free Vtoc Index Records         *
```

Figure 166. Sample JCL for ACBJBAOF (Part 1 of 2)

```
//*    DEVICE     | DEVICETYPE ->    Device Type                     *
//*    DEVNUM     | ADDRESS    ->    Device Number                   *
//*    SHARE      | SHRDASD    ->    Shared DASD                     *
//*    USE        | USEATTR    ->    Use Attributes                  *
//*    RDCACHE    | RDSTAT     ->    RD Cache Status                 *
//*    DASDFW                  ->    DASD FW Status                  *
//*    CACHEFW                 ->    Cache FW Status                 *
//*    DUPLEX     | DUPLEXST   ->    Duplex Status                   *
//*    OTHER      | OTHERDEV   ->    Other Device                    *
//*    SUBSYS     | SUBSYSID   ->    Subsys ID                       *
//*    PHYSTAT    | STATPHYS   ->    Physical Status                 *
//*    STORGRP    | SG         ->    Storage Group Name              *
//*    CFVOLST                 ->    CF Volume Status                *
//*******************************************************************
//GENREP  EXEC ACBJBAOB,
//        PLIB1=SYS1.DGTPLIB,
//        TABL2=userid.TEST.ISPTABL
//ISPFILE DD  DSN=IBMUSER.DASDVOL.REPORT,DISP=OLD
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQVAR1 DASDLIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD  *
VOLSER
INDXSTAT
FRAG
USEATTR
SG
TITLE=STATUS OF VOLUMES THAT ARE SMS MANAGED
/*
//*******************************************************************
//*                                                                 *
//* COPY THE DASD VOLUME REPORT                                     *
//*                                                                 *
//*     SYSUT1 - INPUT (FROM PREVIOUS STEP)                         *
//*     SYSUT2 - OUTPUT                                             *
//*                                                                 *
//*******************************************************************
//REPGEN   EXEC PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.DASDVOL.REPORT,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*
```

Figure 166. Sample JCL for ACBJBAOF (Part 2 of 2)

Generate Report from ISMF-Saved Tape List: ACBQBAR4

ACBQBAR4 is called by SYS1.SACBCNTL member ACBJBAOT to generate a flat file from an ISMF-saved tape table and lists the fields of your choice, in the order you specify.

See Figure 167 on page 397 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR4 EXEC in batch. Each parameter must be on a separate line.

Parameters	Description
CKPTVOL | CKPT	Prints whether this is a checkpoint volume.
COMPTYPE | TYPECOMP	Prints the compaction type.
CRTDTVOL | VOLCRTDT	Prints the volume's creation date.
EXPDTVOL | VOLEXPDT	Prints the volume's expiration date.
LSTENTDT | LASTEJECT	Prints the last eject/entry date.
LSTMNTDT | LASTMOUNT	Prints the volume's last mount date.
LSTWRTDT | LASTWRITE	Prints the volume's last write date.
LIBNAME	Prints the volume's library name.
MEDIA | MEDIATYPE	Prints the volume's media type.
OWNER | OWNERINFO	Prints the volume's owner information.
PAGELENGTH=nn	Sets the page length for the report (default is 60)
RECTECH | TECHREC	Prints the volume's recording technology.
SHELF | SHELFLOC	Prints the volume's shelf location.
SPCLATTR | ATTRSPCL	Prints the volume's special attributes.
STORGRP | SGNAME	Prints the storage group that the tape belongs to.
TITLE=nn	Prints a title for the report. Title does *not* need to be placed in parentheses or quotation marks and cannot expand more than one record in length.
USEATTR | ATTRUSE	Prints the volume use attributes.
VOLERROR | ERRORVOL	Prints the volume's error information.
VOLLOC | LOCVOL	Prints the volume's location.
VOLSER	Prints the volume serial number.
WRTPROT | PROTWRT	Is the volume write protected?

Sample JCL for Generating a Report from ISMF-saved Tape List

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE TAPE VOLUME REPORT FROM A PREVIOUSLY     *
//* SAVED ISMF MOUNTABLE TAPE VOLUME LIST                           *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*      CHANGE PARAMETERS                                          *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* DELETE STEP, TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY *
//*                                                                 *
//*******************************************************************
//DELREP  EXEC  PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
 DELETE IBMUSER.TAPEVOL.REPORT
/*
//*******************************************************************
//*                                                                 *
//* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED   *
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY    *
//*       DATA SET.                                                 *
//*                                                                 *
//*******************************************************************
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD  DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),
//    BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN  DD *
/*
//*******************************************************************
//*                                                                 *
//* REPORT GENERATION STEP                                          *
//*                                                                 *
//*     PARAMETER FOLLOWING ACBQBAR4 - ISMF SAVED LIST NAME (INPUT) *
//*     ISPFILE - TAPE VOLUME REPORT (OUTPUT, FROM ALCISPFL STEP)   *
//*     SYSIN   - KEYWORDS SPECIFYING THE DATA IN THE REPORT        *
//*                                                                 *
//*******************************************************************
//GENREP  EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//ISPFILE DD  DSN=IBMUSER.TAPEVOL.REPORT,DISP=OLD
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR4 TAPELIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD  *
VOLSER
LASTMOUNT
LASTEJECT
USEATTR
STORGRP
LIBNAME
TITLE=STATUS OF TAPES AS ON 06/01/96
/*
```

Figure 167. Sample JCL for ACBJBAOT (Part 1 of 2)

```
//*******************************************************************
//*                                                                 *
//* COPY THE TAPE VOLUME REPORT                                     *
//*                                                                 *
//*   SYSUT1 - INPUT (FROM PREVIOUS STEP)                           *
//*   SYSUT2 - OUTPUT                                               *
//*                                                                 *
//*******************************************************************
//REPGEN EXEC PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.TAPEVOL.REPORT,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*******************************************************************
```

Figure 167. Sample JCL for ACBJBAOT (Part 2 of 2)

Generate Data Set Report from DCOLLECT Data: ACBQBAR7

ACBQBAR7 is called by SYS1.SACBCNTL member ACBJBARD to generate a flat file from DCOLLECT data taken from data set records and lists the fields of your choice, in the order you specify.

See Figure 168 on page 400 for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR7 EXEC in batch. Each parameter must be on a separate line.

Parameters	Description
ALLOCSP | ALLOCSPC	Prints the data sets allocated space
BACKUP | BACKUPDT	Prints the date of the last backup for data set.
BLKUNUSED	Prints the number of unused blocks for the data set.
CHANGE | CHGIND	Is the change indicator on for data set?
CREATE | CREATEDT	Prints the data set's creation date.
DATACLAS | DC	Prints the data class for the data set, if there is one.
DSNAME | DSN	Prints the data set name.
DSNLENGTH=nn	Limits print of data set name to nn characters. Defaults to 44 characters.
DSORG	Prints the DSORG for data set.
ENTRYTYPE	Prints the data set's entry type.
EXPIRE | EXPIREDT	Prints the data set's expiration date.
LASTREF | LASTREFDT	Prints the data set's last reference date.
LRECL	Prints the data set's record length.
MGMTCLS	Prints the management class for the data set, if there is one.
MULTVOL	Is the data set multivolume?
NUMEXT | EXTNUM	Prints the number of extents for data set.
PAGELENGTH=nn	Sets the page length for the report (default is 60).
PDSE	Is the data set a PDSE?
REBLOCK | REBLK	Is the data set reblockable?
RECFM | RECFMT	Prints the data set's record format.
SMS | MANAGED	Is the data set SMS-managed?
STORCLAS | SC	Prints the storage class for data set, if there is one.
STORGRP | SG	Prints the storage group for data set, if there is one.
TITLE=xxx	Prints a title for the report. Title does *not* need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
TOTALS	Specifies whether you want DSN space totals printed for this DCOLLECT data.
USED%	Prints the percentage of used space for the data set.
VOLSEQ	Prints the volume sequence number for data set.
VOLSER	Prints the volume serial of the data set.
VVRCHK | VVR	If the data set is SMS-managed, does it have a valid VVR or NVR?

Sample JCL for Generating Data Set Report from DCOLLECT Data

```
/*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE DATA SET REPORT FROM DCOLLECT DATA       *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*      CHANGE PARAMETERS                                          *
//*                                                                 *
//*******************************************************************
//*******************************************************************
//*                                                                 *
//* DELETE STEP - TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY *
//*                                                                 *
//*******************************************************************
//DELREP    EXEC  PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
 DELETE IBMUSER.DATASET.REPORT
/*
//*******************************************************************
//*                                                                 *
//* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED   *
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY    *
//*       DATA SET.                                                 *
//*                                                                 *
//*******************************************************************
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD  DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
//    BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN  DD *
/*
//*******************************************************************
//*                                                                 *
//* REPORT GENERATION STEP                                          *
//*                                                                 *
//*    DCOLIN  - DCOLLECT DATA (INPUT)                              *
//*    ISPFILE - DATA SET REPORT (OUTPUT, FROM ALCISPFL STEP)       *
//*    SYSIN   - KEY WORDS TO SPECIFY THE DATA IN THE OUTPUT        *
//*                                                                 *
//*   CHOOSE FROM THE FOLLOWING PARAMETERS FOR DATA TO BE LISTED    *
//*   THE ORDER CHOSEN DETERMINES THE ORDER IN THE REPORT           *
//*   THE TOTAL REPORT WIDTH MAY NOT EXCEED 133                     *
//*                                                                 *
//*   PARAMETERS:                                                   *
//*                                                                 *
//*      ALLOCSP | ALLOCSPC  -   ALLOCATED SPACE                    *
//*      BACKUP | LASTBKUP   -   DATA SET'S LAST BACKUP DATE        *
//*      BLKUNUSED           -   BLOCKS UNUSED                      *
//*      CHANGE | CHGIND     -   CHANGE INDICATOR FOR DATA SET      *
//*      CREATE | CREATEDT   -   DATA SET CREATION DATE             *
//*      DATACLAS | DC       -   DATACLAS OF DATA SET               *
//*      DSNAME | DSN        -   DATASETNAME                        *
//*      DSNLENGTH           -   LIMIT DATASET NAME TO THIS NUMBER  *
//*                          -   OF CHARACTERS                      *
```

Figure 168. Sample JCL for ACBJBARD (Part 1 of 2)

```
//*      DSORG               -   DATA SET ORGANIZATION               *
//*      ENTRYTYPE           -   ENTRY TYPE OF DATA SET              *
//*      EXPIRE | EXPIREDT   -   DATA SET EXPIRATION DATE            *
//*      LASTREF | LASTREFDT -   DATA SET LAST REFERENCE DATE        *
//*      LRECL               -   RECORD LENGTH                       *
//*      MGMTCLAS | MC       -   MANAGEMENT CLASS FOR DATA SET       *
//*      MULTVOL             -   IS THE DATA SET MULI-VOLUME?        *
//*      NUMEXT | EXTNUM     -   NUMBER OF EXTENTS                   *
//*      PAGELENGTH          -   NUMBER OF LINES PER PAGE (DEF IS 60)*
//*      PDSE                -   IS THE DATA SET A PDSE?             *
//*      REBLOCK | REBLK     -   IS THE DATA SET REBLOCKABLE?        *
//*      RECFM | RECFMT      -   RECORD FORMAT                       *
//*      SMS | MANAGED       -   IS THE DATA SET MANAGED?            *
//*      STORCLAS | SC       -   STORAGE CLASS FOR DATA SET          *
//*      STORGRP | SG        -   STORAGE GROUP FOR DATA SET          *
//*      TITLE=XXXXX         -   TITLE FOR REPORT                    *
//*      TOTALS              -   PRINT DSN SPACE TOTALS              *
//*      USED%               -   USED SPACE %                        *
//*      VOLSEQ              -   VOLUME SEQUENCE NUMBER              *
//*      VOLSER              -   VOLUME SERIAL                       *
//*      VVRCHK | VVR        -   IF SMS, IS THERE A VVR OR NVR?      *
//*                                                                  *
//*******************************************************************
//GENREP   EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//DCOLIN  DD  DSN=IBMUSER.DCOLLECT.DATA,DISP=SHR
//ISPFILE DD  DSN=IBMUSER.DATASET.REPORT,DISP=OLD
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR7) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//SYSIN DD  *
DSN
VOLSER
ALLOCSP
RECFM
BLKUNUSED
TITLE=DATA SET REPORT FROM DCOLLECT DATA - 06/01/96
TOTALS
BLKSIZE
EXTNUM
DSORG
STORCLAS
/*
//*******************************************************************
//*                                                                 *
//*  COPY THE DATA SET REPORT                                       *
//*                                                                 *
//*     SYSUT1 - INPUT (FROM PREVIOUS STEP)                         *
//*     SYSUT2 - OUTPUT                                             *
//*                                                                 *
//*******************************************************************
//REPGEN EXEC PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.DATASET.REPORT,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*******************************************************************
```

Figure 168. Sample JCL for ACBJBARD (Part 2 of 2)

Generate Model Commands from ISMF-saved List: ACBQBAM1

SYS1.SACBCNTL member ACBJBAM1 calls ACBQBAM1 to create model commands from a saved &ismf; list.

See Figure 169 for the sample JCL and parameters.

Sample JCL for Generating Model Commands from ISMF-saved List

```
//*******************************************************************
//*                                                                 *
//* SAMPLE JCL TO GENERATE MODEL COMMANDS FROM A SAVED ISMF DATA    *
//* SET LIST IN BATCH                                               *
//*                                                                 *
//*   INSTRUCTIONS BEFORE SUBMITTING:                               *
//*                                                                 *
//*      CHANGE JOBCARD                                             *
//*      CHANGE PREFIX                                              *
//*      CHANGE PARAMETERS                                          *
//*                                                                 *
//*   PARAMETERS:                                                   *
//*                                                                 *
//*      PARAMETER FOLLOWING ACBQBAM1 - SAVED ISMF LIST NAME        *
//*                                     DATA SET OR DASD VOLUME LIST *
//*      REST OF THE PARAMETERS - MODEL COMMAND FOR GENERATION      *
//*                                                                 *
//*******************************************************************
//GENCMDS EXEC  ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN  DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAM1 DSNLIST +
DD DSN=/,VOL=SER=@ +
) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*
```

Figure 169. Sample JCL for ACBJBAM1

Creating DFSMS Online Reports

With NaviQuest you can create a report of the contents of an SMS configuration, or the planned changes to a configuration, in the form of a sequential data set. The report can be printed using your choice of tool or utility.

The following reports can be generated:
- Customized data set reports from ISMF saved data set list
- Customized volume reports from ISMF saved DASD volume list
- Customized volume reports from ISMF saved tape volume list
- Customized data set reports from DCOLLECT data
- Customized volume reports from DCOLLECT data
- SMS configuration reports from DCOLLECT data

Note: These reports can also be generated in batch. “Performing Storage Administration Tasks in Batch” on page 343, provides the CLISTs or EXECs required to run these reports in batch.

Before you can generate any of these reports, you must already have created *either* DCOLLECT data or an ISMF table. After creating your base data, you can select the fields that you want included in the report and the order you want them presented.

For example, if you want the data set, percentage used, and expired date fields printed in that particular order, you would enter a value of 1 in the data set field, a value of 2 in the percentage used field, and a value of 3 in the expired date field. You may number as many or as few fields as you need. To generate SMS reports, choose option 5 from the NaviQuest Primary Option Menu.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBSMDP0 ENHANCED ACS MANAGEMENT - NAVIQUEST PRIMARY OPTION MENU
 Enter Selection or Command ===> ____________________________________________

 Select one of the following options and press Enter:

   1  Test Case Generation
   2  ACS Test Listings Comparison
   3  Enhanced ACS Test Listing
   4  Test Case Update with Test Results
   5  SMS Report Generation
   6  Model Commands Generation
   7  Batch Testing/Configuration Management
   X  Exit









 Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 170. NaviQuest Primary Option Menu, ACBSMDP0

Related Reading: For more information on generating saved lists from ISMF tables, refer to *DFSMS: Using ISMF*.

Creating Data Set Reports from Saved ISMF Lists

To generate data set reports, use the “Data Set Report From Saved ISMF List Entry Panel” panel (option 11.5.1). This panel requires you to set up the parameters of the data sets and to define which fields you want in your report.

1. From the “SMS Report Generation Selection Menu” panel (option 11.5), choose “Data Set Report from Saved ISMF List” (option 1).

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBSFLR1             SMS REPORT GENERATION SELECTION MENU
 Enter Selection or Command ===> ____________________________________________

 Select one of the following options and press Enter:

    1  Data Set Report from Saved ISMF List
    2  DASD Volume Report from Saved ISMF List
    3  Tape Volume Report from Saved ISMF List
    4  Data Set Report from DCOLLECT Data (D records)
    5  DASD Volume Report from DCOLLECT Data (V records)
    6  SMS Configuration Report from DCOLLECT Data (DC, SC, MC, SG, BC,
       VL, AG, DR, LB, CN or AI records)









 Use HELP Command for Help; Use END Command to Exit.
```

Figure 171. SMS Report Generation Selection Menu, ACBSFLR1

After choosing option 1, a panel appears from which you can choose from a selection of items that you want to appear in your report.

2. Fill in the top portion of the next screen (Data Set Report From Saved ISMF List Entry Panel) with the name of the data set that you want to hold the output report, whether you want to replace the existing data in the output data set, the name of the previously saved table, number of lines per page you want in the report, and whether you want totals for the allocated and used space attributes.

   Note: The maximum allowed size for reports is 133 columns and for data set names is 44 bytes. If you select the data set name to appear in the report, you can code the Max Length of the DSN Print field to free some of the 44 bytes.

   Coding the minimum number of 11 frees up 33 extra bytes of report columns; however, only the first 11 bytes of the data set name will be included in the report.

3. Fill in as many fields as you want in the order you want them printed for your report.

   Note: Use PF7 (up) and PF8 (down) to scroll backward or forward.

   For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the data set name, the amount of allocated space, the amount of allocated space actually used, block size, and the optimal size.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDRPR5       DATA SET REPORT FROM SAVED ISMF LIST ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate report, specify the following information and press Enter:
   Saved ISMF List  . . . . . . . . . . ________ (Data Set List)
   Data Set to Hold Report
     ===> ____________________________________________________
     Replace Contents if DSN Exists . . _ (Y or N)  Page Length  . . __
   Max Length of DSN Print  . . . . . . __               Totals  . . _  (Y or N)

 Specify fields in numeric order (max width of report is 133 characters):
                                                                   More:     +
                              Length                                Length
   DATA SET NAME  . . . . . 1_ (45)    SEC ALLOC  . . . . . . . __  (9)
   ALLOC SPACE  . . . . . . 2_  (8)    DS ORG . . . . . . . . . __  (6)
   ALLOC USED . . . . . . . 3_  (8)    REC FMT  . . . . . . . . __  (7)
   % NOT USED . . . . . . . __  (8)    RECORD LENGTH  . . . . . __  (6)
   COMPRESSED FORMAT  . . . __  (7)    BLK SZ CI SIZE . . . . . 4_  (8)
   % USER DATA REDUCT . . . __  (8)    OPTIMAL SIZE . . . . . . 5_  (9)

 Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 172. Data Set Report from Saved ISMF List Entry Panel, ACBDRPR5

4. After you have filled in all the information on the first panel, press PF8 to scroll forward. Fill in the remaining required information. To submit the report for processing, press the Enter key. This creates a saved data set that can be browsed.

Creating Volume Reports from Saved ISMF Lists

To generate DASD volume reports, use “DASD Volume Report from Saved ISMF List.” (option 11.5.2).

1. From the “SMS Report Generation Selection Menu” panel (option 11.5), choose “DASD Volume Report from Saved ISMF List.” (option 2).
2. Fill in the data set name, table member name, page length, and totals.

   Indicate whether you want to replace the specified data set if it already exists. If N is specified and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.
3. Select as many fields as you want in the report, using numbers to indicate the order in which you want them printed.

   For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the amount of free space, the fragmentation index, and the device number, in that order.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDRPR7     DASD VOLUME REPORT FROM SAVED ISMF LIST ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate report, specify the following information and press Enter:
   Saved ISMF List  . . . . . . . . . . _______ (DASD Volume List)
   Data Set to Hold Report
     ===> ____________________________________________________
     Replace Contents if DSN Exists . . _ (Y or N)  Page Length  . . __
                                                          Totals  . . _  (Y or N)

 Specify fields in numeric order (max width of report is 133 characters):
                                                                  More:     +
                               Length                                Length
   VOLUME SERIAL  . . . . . 1_  (7)    FREE VIRS  . . . . . . . . __  (9)
   FREE SPACE . . . . . . . 2_  (9)    DEVICE TYPE  . . . . . . __  (8)
   % FREE . . . . . . . . . __  (6)    DEV NUMBER . . . . . . . 4_  (7)
   ALLOC SPACE  . . . . . . __  (9)    SHR DASD . . . . . . . . __  (8)
   FRAG INDEX . . . . . . . 3_  (8)    USE ATTR . . . . . . . . __  (8)
   LARGEST EXTENT . . . . . __  (9)    RD CACHE STATUS  . . . . __  (9)

 Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 173. DASD Volume Report from Saved ISMF List Entry Panel, ACBDRPR7

4. After you have filled in all the information, press the Enter key. This creates a saved data set that can be browsed.

Creating Customized Tape Reports from Saved ISMF Lists

To generate tape reports, use option 11.5.3:

1. From the “SMS Report Generation Selection Menu” panel (option 11.5), choose “Tape Volume Report from Saved ISMF List” (option 3).
2. Fill in the data set name, table member name, and page length.

   Indicate whether you want to replace the specified data set if it already exists. If N is specified and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.
3. Select as many fields as you want in the report, using numbers to indicate the order in which you want them printed.

   For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the use attribute, the volume error status, the media type, and the shelf location.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDFLR3     TAPE VOLUME REPORT FROM SAVED ISMF LIST ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate report, specify the following information and press Enter:
   Saved ISMF List  . . . . . . . . . . _______ (Tape Volume List)
   Data Set to Hold Report
     ===> ____________________________________________________________
     Replace Contents if DSN Exists . . _ (Y or N)  Page Length  . . __

 Specify fields in numeric order (max width of report is 133 characters):
                                                                   More:     +
                                 Length                            Length
   VOLUME SERIAL  . . . . . 1_  (7)    LAST WRITTEN . . . . . . __ (11)
   USE ATTR . . . . . . . . 2_  (8)    LAST MOUNT . . . . . . . __ (11)
   VOL ERROR ST . . . . . . 3_ (19)    LAST ENTER/EJECT . . . . __ (11)
   CKPT VOLUME  . . . . . . __  (5)    VOLUME EXPDT . . . . . . __ (11)
   LIBRARY NAME . . . . . . __  (9)    VOL CREATE DATE  . . . . __ (11)
   STORGRP NAME . . . . . . __  (9)    WRITE PROTECT  . . . . . __  (8)
   MEDIA TYPE . . . . . . . 4_  (7)    VOLUME LOCATION  . . . . 5_  (8)

 Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 174. Tape Volume Report from Saved ISMF List Entry Panel, ACBDFLR3

4. After you have filled in all the information, press the Enter key. This creates a saved data set that can be browsed.

Data Set Report from DCOLLECT Data

To generate data set reports from DCOLLECT data, use option 11.5.4. Select the fields that are to be included in the report with a number to indicate the order they are to be printed.

The filled in fields of the following sample panel indicate that the user wants a report that prints the data set name, the amount of allocated space, the percentage used, the data set organization, and the volume serial number, in that order.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDRPR1        DATA SET REPORT FROM DCOLLECT DATA ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate report, specify the following information and press Enter:
   DCOLLECT Data  . . ____________________________________________________
   Data Set to Hold Report
     ===> ____________________________________________________
     Replace Contents if DSN Exists . . _ (Y or N)  Page Length  . . __
   Max Length of DSN Print  . . . . . . __               Totals  . . _  (Y or N)

 Specify fields in numeric order (max width of report is 133 characters):
                                                                  More:     +
                              Length                             Length
   DATA SET NAME  . . . . . 1_ (45)    CHG IND  . . . . . . . . __  (7)
   ALLOC SPACE  . . . . . . 2_ (10)    DS ORG . . . . . . . . . 4_  (6)
   % USED . . . . . . . . . 3_  (6)    REC FMT  . . . . . . . . __  (7)
   BLOCK UNUSED . . . . . . __  (8)    RECORD LENGTH  . . . . . __  (6)
   BLK SZ CI SIZE . . . . . __  (8)    SMS-MANAGED  . . . . . . __  (5)
   EXPIRE DATE  . . . . . . __ (11)    VOLUME SERIAL  . . . . . 5_  (7)

 Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 175. Data Set Report from DCOLLECT Data Entry Panel, ACBDRPR1

Note: Use PF7 (up) to scroll backward and PF8 (down) to scroll forward. Press the Enter key to generate the report.

DASD Volume Report from DCOLLECT Data

To generate volume reports from DCOLLECT data, use option 11.5.5. Select the fields that are to be included in the report with a number to indicate the order they are to be printed.

For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the index status, the use attribute, and the number of free extents.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDRPR2      DASD VOLUME REPORT FROM DCOLLECT DATA ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate report, specify the following information and press Enter:
   DCOLLECT Data  . . ____________________________________________________
   Data Set to Hold Report
     ===> ____________________________________________________
     Replace Contents if DSN Exists . . _ (Y or N)  Page Length  . . __
                                                    Totals . . . . . _  (Y or N)

 Specify fields in numeric order (max width of report is 133 characters):
                                                                   More:     +
                               Length                             Length
   VOLUME SERIAL  . . . . . 1_  (7)    LARGEST EXTENT . . . . . __  (8)
   INDEX STATUS . . . . . . 2_  (9)    FREE EXTENTS . . . . . . 4_  (8)
   USE ATTRIBUTE  . . . . . 3_  (8)    FREE DSCBS . . . . . . . __  (9)
   SHARED DASD  . . . . . . __  (8)    FREE VIRS  . . . . . . . __  (8)
   % FREESPACE  . . . . . . __  (6)    DEVICE TYPE  . . . . . . __  (8)
   FREESPACE (KB) . . . . . __  (8)    DEV NUMBER . . . . . . . __  (7)

 Use DOWN Command to Scroll Forward; Use UP Command to Scroll Backward;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 176. DASD Volume Report from DCOLLECT Data Entry Panel, ACBDRPR2

SMS Configuration Report from DCOLLECT Data

To generate SMS configuration reports from DCOLLECT data, use option 11.5.6. Select the fields that are to be included in the report with either Yes or No (Y or N).

Y include this record in the report

N do not include this record in the report

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDRPR3   DFSMS CONFIGURATION REPORT FROM DCOLLECT DATA ENTRY PANEL
 Command ===> ____________________________________________________________________

 To generate report, specify the following information and press Enter:
   DCOLLECT Data  . . ____________________________________________________
   Data Set to Hold Report
     ===> ____________________________________________________
     Replace Contents if DSN Exists . . _ (Y or N)  Page Length  . . __

   Specify information to be included in the report: (Y or N)

     Data Classes (DC) . . . . _       Storage Group Volumes (VL) . _
     Storage Classes (SC)  . . _       OAM Drives (DR)  . . . . . . _
     Management Classes (MC) . _       OAM Libraries (LB) . . . . . _
     Storage Groups (SG) . . . _       Cache Names (CN) . . . . . . _
     Base Configuration (BC) . _       ACS Accounting (AI)  . . . . _
     Aggregate Groups (AG) . . _



 Use HELP Command for Help; Use END Command to Exit.
```

Figure 177. SMS Configuration Report from DCOLLECT Data Entry Panel, ACBDRPR3

Additional Storage Administration Functions

The following are additional storage administration functions:

- QSAVE and QRETRIEV ISMF commands

  The QSAVE and QRETRIEV ISMF commands let you save a “query” of frequently used parameters under ISMF. You can then retrieve the parameters by their query names. The QSAVE and QRETRIEV ISMF commands can be used while running the ISMF data set list or volume list options, interactively or in batch.
- Model Command Generator (ACBQFLM1 EXEC for saved ISMF lists and ACBQADM2 EXEC for DCOLLECT data)

  The Model Command Generator option creates a “model” command against each entry in a data set saved ISMF list, a saved ISMF volume list, or from DCOLLECT data. NaviQuest does the symbolic substitution.
- COPYFILT macro

Using the COPYFILT macro you can create synchronized filter lists (FILTLISTs) that can be applied across all ACS routines. Create a FILTLIST member in your ACS routine source data set. Make all filter list updates there. Then call the COPYFILT macro from the command line to have the changes replicated across all of the ACS routines.

QSAVE and QRETRIEV Commands: Saving and Retrieving ISMF Selection Criteria

The QSAVE function saves all the variables of an ISMF query used for a particular run. To use the same variables in a later run, call QRETRIEV from the “Data Set

Selection Entry Panel” or the “Volume Selection Entry Panel”. The QRETRIEV function retrieves all the values used in the first run and reuses them for a batch or an interactive session.

How to Use the QSAVE and QRETRIEV Commands

When a query is saved, a new entry is created in an ISPF table. ISMFQDSN is the table name for data set queries. ISMFQVOL is the table name for volume queries. If the tables do not exist when the first query is saved, the table is created into the library pointed to by DDNAME ISPTABL.

Duplicate query names within the ISMFQDSN and ISMFQVOL tables are not allowed. An error message indicating that there is a duplicate query name will be received if an attempt is made to save a query that already exists. An attempt to retrieve a query that does not exist, will receive an error message indicating that no query has been found.

To use either command, perform the following steps:

1. Create and then save a query.

   The user has two options for creating a query:

   - Create the query while running ISMF in batch.

     Specify the QSAVE(*name*) parameter in the JCL for the batch job. Where *name* is the query name to be saved.
   - Generate the QUERY online.

     To do this, fill in all desired parameters for the query from either ISMF option 1 (data set list option) or option 2 (volume list option). Then return to the first page of the selection panels.

     To save the query, enter the query name on the field marked either QUERY NAME TO SAVE or RETRIEVE. Go to the top of the panel and enter QSAVE. You will receive messages indicating that the query name has been successfully saved.

   The following sample panel, “Data Set Selection Entry Panel”, is the first panel with fields filled in.

```
 Panel  Defaults  Utilities  Scroll  Help
--------------------------------------------------------------------------------
 DGTDDDS1                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)
        2 Generate list from Catalog
           Catalog Name . . .
             Catalog Password . . . .             (if password protected)
           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 178. Data Set Selection Entry Panel, DGTDDDS1

2. After the query is saved, retrieve the query name by specifying the QUERY(name) parameter in a batch job or by filling in the QUERY NAME TO SAVE OR RETRIEVE field on the data set or volume option panel of ISMF. Then issue the QRETRIEV command at the top of the panel.

Creating Model Commands

You can create a user-specified model command to build a set of commands for each data set or volume in the DCOLLECT data or saved ISMF list. The ACBQFLM1 EXEC uses a saved ISMF list and the ACBQADM2 EXEC uses DCOLLECT data.

How to Create Model Commands

To create DFSMSdss commands that delete selected data sets:

1. Use ISMF to build an ISMF table of the data sets you want deleted.
2. Use the model command facility to build your DSS commands.
3. Specify both the ISMF table and the DSS model command you want applied to all the data sets in the list.

To change the management class assigned to a group of data sets of a given high level qualifier whose user requirements have changed (high level qualifier ABC to management class MCSTAND):

1. Use ISMF to create a list of data sets that meet the criteria you want.
2. Tailor the list to remove any data sets that you do not want processed by the model command function.

3. Save the ISMF list into a saved table.

   Example of Saved Data Set List:

   ```
   ABC.BACKUP.LAB.REVIEW
   ABC.BACKUP.LAB.SCRIPT
   ABC.BACKUP.LAB.JCL
   ```

   Note: In steps 1 through 3 of this example, an ISMF-saved list is used as input to generate model commands. The remaining steps (4 through 8) apply to either ISMF-saved lists or DCOLLECT records as model command input.

4. Choose option 11.6. The ISMF " Model Commands Generation Selection Menu" panel appears. See Figure 179.

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBSUTM3           MODEL COMMANDS GENERATION SELECTION MENU
 Enter Selection or Command ===> ____________________________________________

 Select the input data to be used and press Enter:

   1  Saved ISMF List
   2  DCOLLECT Data

 Use HELP Command for Help; Use END Command to Exit.
```

Figure 179. Model Commands Generation Selection Menu, ACBSUTM3

5. Select either option 1 or 2 (11.6.1 or 11.6.2).
6. Fill in the appropriate fields:
   - If you selected option **1** (ACBQFLM1 EXEC), enter the following information in the " Model Commands Generation from Saved ISMF List Entry Panel":
     - In the first field, the name of the table just saved
     - In the second field,
       ALTER / MANAGEMENTCLAS(newmgmtclas)

       Example of Model Command:

       ```
       ALTER / MGMTCLAS(MCSTAND)
       ```

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDUTM1  MODEL COMMANDS GENERATION FROM SAVED ISMF LIST ENTRY PANEL
 Command ===> ____________________________________________________________

 To generate model commands, specify the following information and press Enter:

   Saved ISMF List . . . ________ (Data Set List or DASD Volume List)

   Model Command for Generation
     ===> ________________________________________________________________


 Note: Available substitutions in the Model Command for Generation field
       are '/' for Data Set Name and '@' for Volume Serial.


 Use HELP Command for Help; Use END Command to Exit.
```

Figure 180. Model Commands Generation from Saved ISMF List Entry Panel, ACBDUTM1

- You can also generate model commands with DCOLLECT data as input, instead of using saved lists. To do this, select option **2** (11.6.2) (ACBQADM2 EXEC) and enter the following information in the “Model Commands Generation from DCOLLECT Data Entry Panel”:
  - In the first field, the name of the DCOLLECT data set
  - In the second field,
    ALTER / MANAGEMENTCLAS(newmgmtclas)

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDUTM2   MODEL COMMANDS GENERATION FROM DCOLLECT DATA ENTRY PANEL
 Command ===> ____________________________________________________________

 To generate model commands, specify the following information and press Enter:

   Data Set Containing DCOLLECT Data
     ===> ______________________________________________________________

   Model Command for Generation
     ===> ________________________________________________________________


 Note: Available substitutions in the Model Command for Generation field
       are '/' for Data Set Name and '@' for Volume Serial.


 Use HELP Command for Help; Use END Command to Exit.
```

Figure 181. Model Commands Generation from DCOLLECT Data Entry Panel, ACBDUTM2

The model command function uses the same conventions as ISPF option 3.4. That is, a slash (/) may be located in the command wherever the user wants the data set name substituted. An “at” sign (@) may be used in the command wherever the user wants the volume serial number, which a data set resides on substituted. However, only two slashes and one @ may be used in each model command.

Attention: There is no syntax checking performed for the command entered by the user. It is the user’s responsibility to correctly enter the command to be created against the entries in the table or DCOLLECT data.

Example Output from Command Generation:

```
ALTER ABC.BACKUP.LAB.REVIEW MGMTCLAS(MCSTAND)
ALTER ABC.BACKUP.LAB.SCRIPT MGMTCLAS(MCSTAND)
ALTER ABC.BACKUP.LAB.JCL MGMTCLAS(MCSTAND)
```

This output will be created in a sequential data set with the name USERID.M*xx*.MODELCMD, where *xx* varies from 1 to 99.

7. Add JCL to the commands to complete the job. When the command generation completes, you are placed in ISPF “edit.”

   Example Job:

   ```
   //ALTER JOB ----
   //STEP1 EXEC PGM=IDCAMS
   //SYSPRINT DD SYSOUT=*
   //SYSIN    DD *
         ALTER ABC.BACK.LAB.REVIEW MGMTCLAS(MCSTAND)
         ALTER ABC.BACK.LAB.SCRIPT MGMTCLAS(MCSTAND)
         ALTER ABC.BACK.LAB.JCL MGMTCLAS(MCSTAND)
   /*
   ```

   Note: You can also use the SYSIN DD statement to reference the sequential data set that contains the output commands instead of the SYSIN statement followed by commands as shown above.

8. Submit the job for execution.

The model command facility can also be used with volume lists from ISPF tables or DCOLLECT data. The following example using volume lists creates a job to DEFRAG all volumes in an ISMF volume list:

Example of Saved List:

```
SCR001 -------
SCR002 -------
SCR003 -------
SCR004 -------
SDV000 -------
SDV001 -------
SHR200 -------
```

Example of Model Command:

```
DEFRAG DYNAM (@)
```

Example Output Command Generation:

```
DEFRAG DYNAM(SCR001)
DEFRAG DYNAM(SCR002)
DEFRAG DYNAM(SCR003)
DEFRAG DYNAM(SCR004)
DEFRAG DYNAM(SDV000)
DEFRAG DYNAM(SDV001)
DEFRAG DYNAM(SHR200)
```

Example Job:

```
//DEFRAG JOB  ----
//STEP1 EXEC PGM=ADRDSSU, PARM='TRACE=YES'
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFRAG DYNAM(SCROO1)
  DEFRAG DYNAM(SCR002)
  DEFRAG DYNAM(SCR003)
  DEFRAG DYNAM(SCR004)
  DEFRAG DYNAM(SDV000)
  DEFRAG DYNAM(SDV001)
  DEFRAG DYNAM(SHR200)
/*
```

Note: During the model command generation, only the first two slashes (//) and the first “at” sign (@) will be substituted.

COPYFILT Macro: COPYLIB Facility for FILTLISTs

As you proceed in the DFSMS implementation, data subtypes are gradually moved to SMS management. Often, the only difference between adding one data subtype and the next is adding data to a filter list (FILTLIST) or creating a new FILTLIST. In addition, FILTLISTs must often be added or updated in multiple routines. The COPYFILT facility simplifies this process. COPYFILT is a COPYLIB facility for filter lists contained in ACS routines.

How to Set Up the COPYFILT Macro

Programming Interface Information

To initially setup the COPYFILT macro, the user must perform the following steps:

1. Create one PDS containing all the related ACS source routines for a single SMS configuration.
2. Group all the FILTLISTs in an ACS routine together in one contiguous section of ACS code.
3. Create a new member containing a consolidated list of all FILTLISTs in all related ACS routines.
4. Before using this macro for the first time, some initial “priming” of the FILTLIST member must be performed. This step need only be done *once*; after that, the process is automatic.

   The required priming has the following format:

   - The line just before the first FILTLIST must have data (beginning in column 1) of /* */
   - All FILTLISTs for all ACS routines follow the line of special character strings.
   - The line after the last FILTLIST must have data (beginning in column 1) of /** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ ** _ **/

5. Make sure each ACS routine using the macro (the data class, storage class, management class, and the storage group routines) has the text strings listed above placed in each of the ACS routines at the location you want the COPYFILT macro to place the consolidated filter list. The COPYFILT process copies the FILTLISTs from the FILTLIST member and places them between these two lines of special character strings.

End of Programming Interface Information

How to Use the COPYFILT Macro after Initial Setup

To use the COPYFILT macro *after* you have completed the initial setup, perform the following steps:

1. Change the FILTLIST member that is used by all four ACS routines. To do this, edit the member containing the FILTLISTs. At the command line, save the member containing the FILTLISTs with the SAVE command.
2. Enter COPYFILT at the command line. The user must be in edit mode (that is, ISPF option 2; you cannot use ISPF option 3.4) for a member of the ACS source data set for this facility to work correctly.

   The panel below appears, and the user is then prompted for the following information:

   - Member names of the data class, storage class, management class, and storage group ACS routines inside the data set
   - The member name containing those FILTLISTs
   - A brief description, for change control, of the change made to the FILTLISTs
3. Press the Enter key. The four routines inside the data set are updated automatically, and a message indicates that the update has taken place. If an error occurs, an error message indicates what the problem is.

For example, the fields in the following panel are filled in with typical information:

```
 Panel  Help
-------------------------------------------------------------------------------
 ACBDFIO1                  COPY FILTLISTS ENTRY PANEL
 Command ===>

 To copy FILTLISTs, specify the following information and press Enter:

   Member Containing FILTLISTs . . . . ________

   Data Class ACS Routine Name . . . . DATACLAS
   Storage Class ACS Routine Name  . . STORCLAS
   Management Class ACS Routine Name . MGMTCLAS
   Storage Group ACS Routine Name  . . STORGRP_

   Change Log Entry . . Migrating DB2 tables__________________________









 Use HELP Command for Help; Use END Command to Exit.
```

Figure 182. Copy FILTLISTs Entry Panel, ACBDFIO1

Appendix F. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

© Copyright IBM Corp. 1986, 2002

Notices

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

© Copyright IBM Corp. 1986, 2002

IBM Corporation
Information Enabling Requests
Dept. DZWA
5600 Cottle Road
San Jose, CA 95193 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Programming Interface Information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of DFSMS.

This publication also documents intended Programming Interfaces that allow you to write programs to obtain the services of DFSMS. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming Interface Information

Programming Interface Information

End of Programming Interface Information

Trademarks

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

CICS
DATABASE 2
DB2
DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
Enterprise Storage Server
ESCON
FlashCopy
IBM
IBMLink
MVS
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
OpenEdition
OS/390
Parallel Sysplex
RACF
RAMAC
RMF
VTAM
z/OS

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

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual or view the *IBM Dictionary of Computing Terms* located at:

http://www.ibm.com/networking/nsg/nsgmain.htm

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

See: This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

A

access method services. A multifunction service program that manages VSAM and non-VSAM data sets, as well as integrated catalog facility (ICF) catalog. Access method services provides the following functions:

- defines and allocates space for data sets and catalogs
- converts indexed-sequential data sets to key-sequenced data sets
- modifies data set attributes in the catalog
- reorganizes data sets
- facilitates data portability among operating systems
- creates backup copies of data sets
- assists in making inaccessible data sets accessible
- lists the records of data sets and catalogs
- defines and builds alternate indexes
- converts CVOLS to ICF catalogs

ACDS. See *active control data set.*

ACS installation exit. User-written code, run after an ACS routine, that provides capabilities beyond the scope of the ACS routine.

ACS interface routine. This calls an ACS routine from an ACS installation-exit routine.

ACS. See *automatic class selection.*

activate. To load the contents of an SCDS into SMS address space storage and into an ACDS or to load the contents of an existing ACDS into SMS address space storage. This establishes a new storage management policy for the SMS complex.

active configuration. The configuration currently used by SMS to control the managed storage in the installation.

active control data set (ACDS). A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACDS is shared by each system that is using the same SMS configuration to manage storage. See also *source control data set, communications data set*.

active data. (1) (1) Data that can be accessed without any special action by the user, such as data on primary storage or migrated data. Active data also can be stored on tape volumes. (2) (2) For tape mount management, application data that is frequently referenced, small in size, and managed better on DASD than on tape. (3) Contrast with *inactive data*.

aggregate backup. The process of copying an aggregate group and recovery instructions so that a collection of data sets can be recovered later as a group.

© Copyright IBM Corp. 1986, 2002

aggregate group. A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy.

ATL. Automated tape library.

ATLDS. Automated tape library dataserver.

automated tape library data server. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. Contrast with *manual tape library*. See also *tape library*.

automatic backup. (1) (1) In DFSMShsm, the process of automatically copying data sets from primary storage volumes or migration volumes to backup volumes. (2) (2) In OAM, the process of automatically copying a primary copy of an object from DASD, optical, or tape volume to a backup volume contained in an object backup storage group.

automatic class selection (ACS) routine. A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

automatic dump. In DFSMShsm, the process of using DFSMSdss automatically to do a full-volume dump of all allocated space on a primary storage volume to designated tape dump volumes.

automatic primary space management insert. In DFSMShsm, the process of deleting expired data sets, deleting temporary data sets, releasing unused space, and migrating data sets from primary storage volumes automatically.

automatic secondary space management. In DFSMShsm, the process of automatically deleting expired migrated data sets, deleting expired records from the migration control data sets, and migrating eligible data sets from migration level 1 volumes to migration level 2 volumes.

availability. For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

B

backup. The process of creating a copy of a data set or object to be used in case of accidental loss.

backup control data set (BCDS). In DFSMShsm, a VSAM key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMShsm.

base configuration. The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration manages.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. See also *integrated catalog facility catalog*.

BCDS. See *Backup control data set.*

BCS. See *Basic catalog structure.*

C

cache fast write. A storage control capability in which the data is written directly to cache without using nonvolatile storage. Cache fast write is useful for temporary data or data that is readily recreated, such as the sort work files created by DFSORT. Contrast with *DASD fast write*.

cache set. A parameter on storage class and defined in the base configuration information that maps a logical name to a set of CF cache structure names.

capacity planning. The process of forecasting and calculating the appropriate amount of physical computing resources required to accommodate an expected workload.

CCSID. See *coded character set identifier*.

CDRA. See *character data representation architecture API*.

CDS. See *Control data set.*

CF. See *coupling facility*.

Channel-to-channel. A method of connecting two computing devices.

Character Data Representation Architecture (CDRA) API. A set of identifiers, services, supporting resources, and conventions for consistent representation, processing, and interchange of character data.

CICS. Customer Information Control System.

class transition. An event that brings about change to an object's service-level criteria, causing OAM to invoke ACS routines to assign a new storage class or management class to the object.

cluster. A data component and an index component in a VSAM key-sequenced data set; or a data component alone in a VSAM entry-sequenced data set. See also *storage cluster.*

Coded Character Set Identifier (CCSID). A 16-bit number that identifies a specific encoding scheme identifier, character set identifiers, code page identifiers, and additional coding required information. The CCSID uniquely identifies the coded graphic character representation used.

collection. A group of objects that typically have similar performance, availability, backup, retention, and class transition characteristics. A collection is used to catalog a large number of objects which, if cataloged separately, could require an extremely large catalog.

COMMDS. Communication data set.

communications data set (COMMDS). The primary means of communication among systems governed by a single SMS configuration. The COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume, which helps balance space among systems running SMS. See also *active control data set* and *source control data set*.

compatibility mode. The mode of running SMS in which no more than eight names—representing systems, system groups, or both—are supported in the SMS configuration.

compress. (1) (1) To reduce the amount of storage required for a given data set by having the system replace identical words or phrases with a shorter token associated with the word or phrase. (2) (2) To reclaim the unused and unavailable space in a partitioned data set that results from deleting or modifying members by moving all unused space to the end of the data set.

compressed format. A particular type of extended-format data set specified with the (COMPACTION) parameter of data class. VSAM can compress individual records in a compressed-format data set. SAM can compress individual blocks in a compressed-format data set. See *compress*.

concurrent copy. A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

connectivity. (1) (1) The considerations regarding how storage controls are joined to DASD and processors to achieve adequate data paths (and alternative data paths) to meet data availability needs. (2) (2) In a system-managed storage environment, the system status of volumes and storage groups.

construct. One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

control data set (CDS). (1) With respect to SMS, a VSAM linear data set containing configurational, operational, or communication information. SMS uses three types of control data sets: the source control data set (SCDS), the active control data set (ACDS), and the communication data set (COMMDS). (2) In DFSMSrmm, all changes made to the inventory, such as adding or deleting volumes, are recorded in the control data set.

CONVERT. A physical volume status indicating all the data sets on a given volume have an associated storage class and are not cataloged in an integrated catalog facility catalog. SMS can select a CONVERT volume for all supported functions. CONVERT volumes are sometimes referred to as converted. See also *INITIAL*.

convert in place. See *in-place conversion*.

coupling facility (CF). The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

coupling facility (CF) cache structure. The CF hardware that provides a data cache.

coupling facility (CF) lock structure. The CF hardware that supports sysplex-wide locking.

Cross-system coupling facility (XCF). A component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

CTC. Channel-to-channel.

D

DADSM. See *direct access device space management*.

DASD. See *direct access storage device*.

DASD fast write. An extended function of some models of the IBM 3990 Storage Control in which data is written concurrently to cache and non-volatile storage and automatically scheduled for destaging to DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. Use of DASD fast write for system-managed data sets is controlled by storage class attributes to improve performance. See also *dynamic cache management*. Contrast with *cache fast write*.

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also *volume, primary storage, migration level 1, migration level 2*.

DATABASE 2™ (DB2®). An IBM relational database management system that is available as a licensed program on several operating systems. Programmers and users of DB2 can create, access, modify, and delete data in relational tables using a variety of interfaces.

data class. A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

data collection application. An ISMF application that allows the storage administrator interactively to submit IDCAMS DCOLLECT background jobs or save JCL for later use.

Data Facility Sort (DFSORT). An IBM licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copying operations, as well as providing versatile data manipulation at the record, field, and bit level.

Data Facility Storage Management Subsystem (DFSMS). An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

data set. In DFSMS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*. In z/OS UNIX environments, the terms *data set* and *file* have quite distinct meanings.

data set collection. A group of data sets which are intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking.

data set stacking. The function used to place several data sets on the same tape volume or set of tape volumes. It increases the efficiency of tape media usage and reduces the overall number of tape volumes needed by allocation. It also allows an installation to group related data sets together on a minimum number of tape volumes, which is useful when sending data offsite.

DB2. See *DATABASE 2*.

DCME. See *dynamic cache management*.

DDM. See *Distributed Data Management Architecture*.

default device geometry. Part of the SMS base configuration, it identifies the number of bytes per track and the number of tracks per cylinder for converting space requests made in tracks or cylinders into bytes, when no unit name has been specified.

default management class. Part of the SMS base configuration, it identifies the management class that should be used for system-managed data sets that do not have a management class assigned.

default unit. Part of the SMS base configuration, it identifies an esoteric (such as SYSDA) or generic (such as 3390) device name. If a user omits the UNIT parameter on the JCL or the dynamic allocation equivalent, SMS applies the default unit if the data set has a disposition of MOD or NEW and is *not* system-managed.

device category. A storage device classification used by SMS. The device categories are as follows SMS-managed DASD, SMS-managed tape, non-SMS-managed DASD non-SMS-managed tape.

Device Support Facilities (ICKDSF). A program used for initialization of DASD volumes and track recovery.

DFSMSdfp. A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMShsm control data set. In DFSMShsm, one of three VSAM key-sequenced data sets that contain records used in DFSMShsm processing. See also *backup control data set, migration control data set, offline control data set*.

DFSMS Network File System. See *z/OS Network File System*.

DFSMSrmm. A DFSMS functional component or base element of z/OS, that manages removable media.

DFSORT. See *Data Facility Sort*.

direct access device space management (DADSM). A collection of subroutines that manages space on disk volumes. The subroutines are: Create, Scratch, Extend, and Partial Release.

direct access storage device (DASD). A mass storage medium on which a computer stores data. Contrast with *random access memory*.

DISALL (disable all). Relationship that prevents a system from allocating or accessing data sets in a VIO storage group, a pool storage group, object or object backup storage group, or individual volumes within a pool storage group.

DISNEW (disable new). Relationship that prevents a system from allocating new data sets in a VIO storage

group, a pool storage group, object or object backup storage group, or individual volumes within a pool storage group.

Distributed Data Management Architecture (DDM). Distributed Data Management Architecture (DDM) offers a vocabulary and a set of rules for sharing and accessing data among like and unlike computer systems. DDM includes a set of standardized file models for keyed, relative record, sequential, and stream data. It allows users and applications to access data without concern for the location or format of the data.

drive definition. A set of attributes used to define an optical disk drive as a member of a real optical library or pseudo optical library.

dual copy. A high availability function made possible by nonvolatile storage in some models of the IBM 3990 Storage Control. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical 3990 subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

dummy storage group. A type of storage group that contains the serial numbers of volumes no longer connected to a system. Dummy storage groups allow existing JCL to function without having to be changed. See also *storage group*.

dump class. A set of characteristics that describes how volume dumps are managed by DFSMShsm.

duplexing. The process of writing two sets of identical records in order to create a second copy of data.

dynamic cache management. A function that automatically determines which data sets will be cached based on the 3990 subsystem load, the characteristics of the data set, and the performance requirements defined by the storage administrator.

E

ENABLE. Relationship that allows a system to allocate and access data sets in a VIO storage group, a pool storage group, or individual volumes within a pool storage group.

Enterprise Storage Server (ESS). An IBM high-end storage subsystem succeeding the 3880 family, the 3990 family, and the 9340 family. Designed for mid-range and high-end environments, the ESS provides you with large capacity, high performance, continuous availability, and storage expandability.

ESS. See *Enterprise Storage Server*.

expiration. (1) The process by which data sets or objects are identified for deletion because their expiration date or retention period has passed. On DASD, data sets and objects are deleted. On tape, when all data sets have reached their expiration date, the tape volume is available for reuse. (2) In DFSMSrmm, all volumes have an expiration date or retention period set for them either by vital record specification policy, by user-specified JCL when writing a data set to the volume, or by an installation default. When a volume reaches its expiration date or retention period, it becomes eligible for release.

extended addressability. The ability to create and access a VSAM data set that is greater than 4 GB in size. Extended addressability data sets must be allocated with DSNTYPE=EXT and EXTENDED ADDRESSABILITY=Y.

extended format. The format of a data set that has a data set name type (DSNTYPE) of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different. Data sets in extended format can be striped or compressed. Data in an extended format VSAM KSDS can be compressed. See also *striped data set, compressed format*.

extended remote copy (XRC). A technique involving both the DFSMS host and the I/O Subsystem that keeps a ″real time″ copy of designated data at another location. Updates to the primary center are replicated at the secondary center asynchronously.

F

file. A collection of information treated as a unit. In z/OS non-UNIX environments, the terms *data set* and *file* are generally equivalent and are sometimes used interchangeably. See also *data set*.

file system. In the z/OS UNIX HFS environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See also *HFS data set*.

FlashCopy. An ESS copy service that provides the appearance of instantaneous replication of a range of track images.

G

giga (G). The information-industry meaning depends upon the context:

1. G = 1,073,741,824(2^{30}) for real and virtual storage
2. G = 1,000,000,000 for disk storage capacity (e.g. 4 GB fixed disk)
3. G = 1,000,000,000 for transmission rates

global resource serialization (GRS). A component of z/OS used for serializing use of system resources and for converting hardware reserves on DASD volumes to data set enqueues.

group. (1) (1) With respect to partitioned data sets, a member and the member's aliases that exist in a PDS or PDSE, or in an unloaded PDSE. (2) (2) A collection of users who can share access authorities for protected resources.

GRS. See *global resource serialization*.

guaranteed space. A storage class attribute indicating that space is to be pre-allocated when a data set is allocated. If explicit volume serial numbers are specified, SMS honors them. If space to satisfy the allocation is not available on the user-specified volumes, the allocation fails.

H

hardware configuration definition (HCD). An interactive interface in MVS that enables an installation to define hardware configurations from a single point of control.

HCD. See *hardware configuration definition*.

HFS. See *hierarchical file system*.

hierarchical file system (HFS) data set. A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using z/OS UNIX System Services. See also *file system.*

HSM complex (HSMplex). One or more z/OS images running DFSMShsm that share a common set of control data sets (MCDS, BCDS, OCDS, and Journal).

I

IDRC. See *improved data recording capability*.

IML. See *Initial microcode load.*

improved data recording capability (IDRC). A recording mode that can increase the effective cartridge data capacity and the effective data rate when enabled and used. IDRC is always enabled on the 3490E Magnetic Tape Subsystem.

inactive data. (1) (1) A copy of active data, such as vital records or a backup copy of a data set. Inactive data is never changed, but can be deleted or superseded by another copy. (2) (2) In tape mount management, data that is written once and never used again. The majority of this data is point-in-time backups. (3) (3) Objects infrequently accessed by users and eligible to be moved to the optical library or shelf. (4) Contrast with *active data*.

index. A set of pointers that are logically ordered by the values of a key. Indexes provide quick access to data and can enforce uniqueness on the rows in a DB2 storage table.

indexed VTOC. A volume table of contents with an index that contains a list of data set names and free space information, which allows data sets to be located more efficiently.

INITIAL. A physical volume status indicating that some data sets on a given volume lack an associated storage class or are not cataloged in an integrated catalog facility catalog. An INITIAL volume is only partially converted to SMS.

Initial Access Response Seconds (IARS). A parameter specified in the definition of an SMS storage class indicating the desired response time to locate, mount, and prepare a piece of media for data transfer.

initial microcode load (IML). The act of loading microcode.

in-place conversion. The process of bringing a volume and the data sets it contains under the control of SMS without data movement, using DFSMSdss.

installation exit. The means specifically described in an IBM software product's documentation by which an IBM software product may be modified by a customer's system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or subroutines to an IBM software product, for the purpose of modifying (including extending) the functions of the IBM software product.

integrated catalog facility catalog. A catalog that is composed of a basic catalog structure (BCS) and its related volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). See also *basic catalog structure, VSAM volume data set*.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS that allows users and storage administrators access to the storage management functions.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive interface for all storage management functions.

interval migration. In DFSMShsm, automatic migration that occurs when a threshold level of occupancy is reached or exceeded on a DFSMShsm-managed volume, during a specified time interval. Data sets are moved from the volume, largest eligible data set first, until the low threshold of occupancy is reached.

invalid. A status indicating that an SCDS or ACDS is in error. Either the SCDS is incomplete, or an ACS routine in the SCDS has assigned an undefined SMS class name or assigned a storage group list containing undefined names. To be complete, an SCDS must contain at least one storage class definition, at least one pool storage group definition containing at least one volume, a storage group selection routine, and base configuration information. An invalid SCDS cannot be activated.

ISMF. See *Interactive Storage Management Facility*.

ISPF. See *Interactive System Productivity Facility*.

J

JCL. See *job control language*.

job control language (JCL). A control language used to identify a job to an operating system and to describe the job's requirements.

K

kilo (K). The information-industry meaning depends upon the context:

1. K = 1024(2^{10}) for real and virtual storage
2. K = 1000 for disk storage capacity (e.g. 4000 KB fixed disk)
3. K = 1000 for transmission rates

L

logical subsystem (LSS). The logical functions of a storage controller that allow one or more host I/O interfaces to access a set of devices. The controller aggregates the devices according to the addressing mechanisms of the associated I/O interfaces. One or more logical subsystems exist on a storage controller. In general, the controller associates a given set of devices with only one logical subsystem.

LSS. See *logical subsystem*.

M

management class. A named collection of management attributes describing the retention and backup characteristics for a group of data sets or for a group of objects in an object storage hierarchy. For objects, the described characteristics also include class transition.

manual tape library. Installation-defined set of tape drives defined as a logical unit together with the set of system-managed volumes which can be mounted on the drives.

MCDS. See *migration control data set*.

mega (M). The information-industry meaning depends upon the context:

1. M = 1,048,576(2^{20}) for real and virtual storage
2. M = 1,000,000 for disk storage capacity (e.g. 4000 MB fixed disk)
3. M = 1,000,000 for transmission rates

MEDIA2. Enhanced Capacity Cartridge System Tape

MEDIA3. High Performance Cartridge Tape

MEDIA4. Extended High Performance Cartridge Tape

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled. See also *migration level 1, migration level 2*.

migration control data set (MCDS). In DFSMShsm, a VSAM key-sequenced data set that contains statistics records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMShsm.

migration level 1. DFSMShsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage, migration level 2*.

migration level 2. DFSMShsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage, migration level 1*.

MTL. Manual Tape Library.

Multiple Virtual Storage (MVS). The short name for MVS/XA™, MVS/ESA, or the MVS element of the z/OS operating system.

MVS. See *Multiple Virtual Storage*.

MVS configuration program (MVSCP). A single-step, batch program that defines the input/output configuration to z/OS.

MVSCP. See *MVS configuration program*.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture. A z/OS operating system environment that supports ESA/390.

MVS/ESA SP. An IBM licensed program used to control the z/OS operating system. MVS/ESA SP together with DFSMS compose the base MVS/ESA operating environment. See also *z/OS*.

MVS system. An MVS image together with its associated hardware, which collectively are often referred to simply as a system, or MVS system.

N

NaviQuest. A component of DFSMSdfp for implementing, verifying, and maintaining your DFSMS SMS environment in batch mode. It provides batch testing and reporting capabilities that can be used to automatically create test cases in bulk, run many other storage management tasks in batch mode, and use supplied ACS code fragments as models when creating your own ACS routines.

nonvolatile storage (NVS). Additional random access electronic storage with a backup battery power source, available with an IBM Cache Storage Control, used to retain data during a power outage. Nonvolatile storage, accessible from all storage directors, stores data during DASD fast write and dual copy operations.

NOTCON (not connected). Relationship that indicates a system is defined but has no access to a VIO storage group, a pool storage group, or individual volumes within a pool storage group.

NVS. See *nonvolatile storage*.

O

OAM. See *object access method*.

OAM complex (OAMplex). One or more instances of OAM running on systems that are part of a parallel sysplex. The OAM systems that are part of an OAM complex share a common OAM database in a DB2 data-sharing group.

OAM-managed volumes. Optical or tape volumes controlled by the object access method (OAM).

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

object backup storage group. A type of storage group that contains optical or tape volumes used for backup copies of objects. See also *storage group*.

object directory tables. A collection of DB2 tables that contain information about the objects that have been stored in an object storage group.

object storage group. A type of storage group that contains objects on DASD, tape, or optical volumes. See also *storage group*.

object storage hierarchy. A hierarchy consisting of objects stored in DB2 table spaces on DASD, on optical or tape volumes that reside in a library, and on optical or tape volumes that reside on a shelf. See also *storage hierarchy.*

object storage tables. A collection of DB2 tables that contain objects.

OpenEdition MVS. See *z/OS UNIX System Services.*

optical disk drive. The mechanism used to seek, read, and write data on an optical disk. An optical disk drive can be operator-accessible or library-resident.

optical library. A storage device that houses optical drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives.

optical volume. Storage space on an optical disk, identified by a volume label. See also *volume*.

OS/390. A network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system. See also *MVS/ESA SP*.

overflow storage group. An SMS storage group used to satisfy allocations which do not fit into the primary storage group. Also known as *spill storage group.*

P

Parallel Sysplex. A collection of MVS systems in a multi-system environment supported by the cross-system coupling facility.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

PDS. See *partitioned data set*.

PDSE. See *partitioned data set extended*.

peer-to-peer remote copy (PPRC). A hardware-based remote copy option that provides a synchronous volume copy across 3990 Model 6 storage subsystems for disaster recovery, device migration, and workload migration.

performance. (1) (1) A measurement of the amount of work a product can produce with a given amount of resources. (2) (2) In a system-managed storage environment, a measurement of effective data

processing speed with respect to objectives set by the storage administrator. Performance is largely determined by throughput, response time, and system availability.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with *temporary data set*.

physical storage. With respect to data, the actual space on a storage device that is to contain data.

physical volume status. Relationship between a volume and SMS. Physical volume status indicates if a volume is SMS-managed, if all of its data sets have an associated storage class, and if all of its data sets are cataloged in an integrated catalog facility catalog. A volume can be in one of three states: CONVERT (fully converted), INITIAL (partially converted), or non-SMS (unconverted).

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

PPRC. See *peer-to-peer remote copy*.

primary data set. When referring to an entire data set collection, the primary data set is the first data set allocated. For individual data sets being stacked, the primary data set is the one in the data set collection that precedes the data set being stacked and is allocated closest to it.

primary space allocation. Amount of space requested by a user for a data set when it is created. Contrast with *secondary space allocation*.

primary storage. A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1, migration level 2*.

primary volume. A volume managed by DFSMShsm containing data sets that are directly accessible to the user

program management. The task of preparing programs for execution, storing the programs, load modules, or program objects in program libraries, and executing them on the operating system.

program object. All or part of a computer program in a form suitable for loading into virtual storage for execution. Program objects are stored in PDSE program libraries and have fewer restrictions than load modules. Program objects are produced by the binder.

pseudo optical library. A set of shelf-resident optical volumes associated with an operator-accessible optical disk drive; see also *real optical library*.

Q

QUIALL (quiesce all). Relationship that prevents a system from scheduling jobs that allocate or access data sets in a VIO storage group, a pool storage group, or individual volumes within a pool storage group.

QUINEW (quiesce new). Relationship that prevents a system from scheduling jobs that allocate new data sets or modify existing ones in a VIO storage group, a pool storage group, or individual volumes within a pool storage group.

R

RACF. See *Resource Access Control Facility*.

RAID. See *redundant array of independent disks*.

RAM. See *random access memory*.

random access memory (RAM). A temporary storage location in which the central processing unit stores and executes its processes. Contrast with direct access storage device (DASD).

read-only variable. An ACS language variable that contains data set or system-derived information. It can be referenced but not altered in an ACS routine.

read-write variable. An ACS language variable that is assigned a value within an ACS routine. It can be referenced, and each ACS routine assigns a value to its own, unique read-write variable.

real optical library. Physical storage device that houses optical disk drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives. Contrast with *pseudo optical library*.

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

redundant array of independent disks (RAID). A direct access storage architecture where data is recorded across multiple physical disks with parity separately recorded so that no loss of access to data results from the loss of any one disk in the array.

Resource Access Control Facility (RACF). An IBM licensed program that is included in z/OS Security Server and is also available as a separate program for the z/OS and VM environments. RACF provides access control by identifying and verifying the users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

Resource Measurement Facility (RMF). An IBM licensed program or optional element of z/OS, that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems.

retained lock. A lock protecting transaction updates when a problem delays transaction recovery of the updates. The retained status is cleared when transaction recovery completes.

RMF. See *resource measurement facility*.

S

SCDS. See *source control data set*.

SCSI. See *small computer system interface*.

SDR. See *sustained data rate*.

secondary space allocation. Amount of additional space requested by the user for a data set when primary space is full. Contrast with *primary space allocation*.

sequential data striping. A software implementation of a disk array that distributes data sets across multiple volumes to improve performance.

sharing control data set. A VSAM linear data set that contains information DFSMSdfp needs to ensure the integrity of the data sharing environment.

SHCDS. See *sharing control data sets*.

shelf. A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

shelf location. (1) A single space on a shelf for storage of removable media. (2) In DFSMSrmm, a shelf location is defined in the removable media library by a rack number, and in a storage location, it is defined by a bin number. See also *rack number, bin number*.

small computer system interface (SCSI). A mechanical, electrical, and functional standard for a small computer input/output bus and command sets for peripheral devices commonly used with small computers.

SMF. See *system management facilities*.

SMS. See *Storage Management Subsystem*.

SMS class. A list of attributes that SMS applies to data sets and objects having similar allocation (data class), performance (storage class), or backup and retention (management class) needs.

SMS complex. A collection of systems or system groups that share a common configuration. All systems in an SMS complex share a common active control data set (ACDS) and a communications data set (COMMDS). The systems or system groups that share the configuration are defined to SMS in the SMS base configuration.

SMS configuration. A configuration base, Storage Management Subsystem class, group, library, and drive definitions, and ACS routines that the Storage Management Subsystem uses to manage storage. See also *configuration, base configuration, source control data set* .

SMS control data set. A VSAM linear data set containing configurational, operational, or communications information that guides the execution of the Storage Management Subsystem. See also *source control data set, active control data set, communications data set*.

SMS-managed data set. A data set that has been assigned a storage class. Also called *system-managed data set*.

SMS-managed volume. A volume, managed by SMS, that is defined in the active configuration. See *CONVERT*.

source control data set (SCDS). A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated using ISMF. See also *active control data set, communications data set*.

spill storage group. See *overflow storage group.*

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage control. The component in a storage subsystem that handles interaction between processor channel and storage devices, runs channel commands, and controls storage devices.

storage director. In a 3990 Storage Control, a logical entity consisting of one or more physical storage paths in the same storage cluster. In a 3880, a storage director is equivalent to a storage path.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD, optical, or tape volumes treated as a single object storage hierarchy. See also

VIO storage group, pool storage group, tape storage group, object storage group, object backup storage group, dummy storage group.

storage group category. A grouping of specific storage groups which contain the same type of data. This concept is analogous to storage pools in a non-system-managed environment.

storage hierarchy. An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, DASD cache), primary storage (DASD containing uncompressed data), migration level 1 (DASD containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format). See also *primary storage, migration level 1, migration level 2, object storage hierarchy*.

storage management. The activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also *Storage Management Subsystem*.

storage management cycle. An invocation of the OAM Storage Management Component (OSMC). The purpose of the storage management cycle is to ensure that every object scheduled for processing is placed in the correct level of the object storage hierarchy (as specified by its storage class), is expired or backed up (as specified by its management class or by an explicit application request), and, if necessary, is flagged for action during a subsequent storage management cycle.

Storage Management Subsystem (SMS). A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

storage subsystem. A storage control and its attached storage devices. See also *tape subsystem*.

stripe. In DFSMS, the portion of a striped data set, such as an extended format data set, that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striped data set. In DFSMS, an extended-format data set consisting of two or more stripes. SMS determines the number of stripes to use based on the value of the SUSTAINED DATA RATE in the storage class. Striped data sets can take advantage of the sequential data striping access technique. See*stripe, striping*.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

sustained data rate. A parameter specified in the definition of an SMS storage class indicating the desired sustained data rate to read the next 4K block of a data entity assuming the prior 4K block has been read.

synchronization time interval. The number of seconds that SMS allows before it checks the COMMDS for volume status, space statistics, and configurational changes from other systems in the complex.

SYSRES. See *system residence volume*.

system data. The data sets required by z/OS or its subsystems for initialization and control.

system group. All systems that are part of the same Parallel Sysplex and are running the Storage Management Subsystem with the same configuration, minus any systems in the Parallel Sysplex that are explicitly defined in the SMS configuration.

system-managed buffering for VSAM. A facility available for system-managed extended-format VSAM data sets in which DFSMSdfp determines the type of buffer management technique along with the number of buffers to use, based on data set and application specifications.

system-managed data set. A data set that has been assigned a storage class.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications. See also *system-managed storage environment*.

system-managed storage environment. An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the system-managed storage environment for z/OS, the function is provided by DFSORT, RACF, and the combination of DFSMS and z/OS.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also *tape library.*

system-managed volume. A DASD, optical, or tape volume that belongs to a storage group. Contrast with *DFSMShsm-managed volume, DFSMSrmm-managed volume*.

system management facilities (SMF). A component of z/OS that collects input/output (I/O) statistics, provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

system name. The name of the system where an SMS operation is being performed.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

system residence (SYSRES) volume. The volume on which the nucleus of the operating system and the master catalog are stored.

T

tape configuration database. One or more volume catalogs used to maintain records of system-managed tape libraries and tape volumes.

tape library. A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives. See also *system-managed tape library, automated tape library data server*.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also *storage group*.

tape subsystem. A magnetic tape subsystem consisting of a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with *permanent data set*.

threshold. A storage group attribute that controls the space usage on DASD volumes, as a percentage of occupied tracks versus total tracks. The *low migration threshold* is used during primary space management and interval migration to determine when to stop processing data. The *high allocation threshold* is used to determine candidate volumes for new data set allocations. Volumes with occupancy lower than the high threshold are selected over volumes that meet or exceed the high threshold value.

translate. To check ACS routine source code for syntactic and semantic errors. If no errors exist, the translation process generates an object table from the source code and places the object table into a specified SCDS.

U

unit affinity. Requests that the system allocate different data sets residing on different removable volumes to the same device during execution of the step to reduce the total number of tape drives required to execute the step. Explicit unit affinity is specified by coding the UNIT=AFF JCL keyword on a DD statement. Implicit unit affinity exists when a DD statement requests more volumes than devices.

use attribute. (1) (1) The attribute assigned to a DAD volume that controls when the volume can be used to allocate new data sets; use attributes are *public*, *private*, and *storage*. (2) (2) For system-managed tape volumes, use attributes are *scratch* and *private*.

user group. A group of users in an installation who represent a single department or function within the organization.

V

validate. To check the completeness and consistency of an individual ACS routine or an entire SMS configuration.

VIO. See *virtual input/output storage group*.

virtual input/output (VIO) storage group. A type of storage group that allocates data sets to paging storage, which simulates a DASD volume. VIO storage groups do not contain any actual DASD volumes. See also *storage group*.

vital records. A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement. Compare with *disaster recovery*.

vital record specification. Policies defined to manage the retention and movement of data sets and volumes for disaster recovery and vital records purposes.

volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume, optical volume, tape volume*.

volume mount analyzer. A program that helps you analyze your current tape environment. With tape mount management, you can identify data sets that can be redirected to the DASD buffer for management using SMS facilities.

volume status. In the Storage Management Subsystem, indicates whether the volume is fully available for system management:

- “Initial” indicates that the volume is not ready for system management because it contains data sets that are ineligible for system management.
- “Converted” indicates that all of the data sets on a volume have an associated storage class and are cataloged in an integrated catalog facility catalog.
- “Non-system-managed” indicates that the volume does not contain any system-managed data sets and has not been initialized as system-managed.

volume table of contents (VTOC).

1. A table on a direct access volume that describes each data set on the volume.
2. An area on a disk or diskette that describes the location, size, and other characteristics of each file and library on the disk or diskette.

VRS. See *vital record specification*.

VSAM record-level sharing (VSAM RLS). An extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. Record-level sharing uses the z/OS Coupling Facility to provide cross-system locking, local buffer invalidation, and cross-system data caching.

VSAM RLS. See *VSAM record-level sharing*.

VSAM sphere. The base cluster of a VSAM data set and its associated alternate indexes.

VSAM volume data set (VVDS). A data set that describes the characteristics of VSAM and system-managed data sets residing on a given DASD volume; part of an integrated catalog facility catalog. See also *basic catalog structure, integrated catalog facility catalog*.

VTOC. See *volume table of contents.*

VVDS. See *VSAM volume data set*.

W

write-once. Describes an optical medium where data can be written one time only.

X

XCF. See *cross-system coupling facility*.

XRC. See *extended remote copy*.

Z

z/OS. z/OS is a network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system.

z/OS Network File System. A base element of z/OS, that allows remote access to z/OS host processor data from workstations, personal computers, or any other system on a TCP/IP network that is using client software for the Network File System protocol.

z/OS UNIX System Services (z/OS UNIX). The set of functions provided by the SHELL and UTILITIES, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the z/OS operating system that allow users to write and run application programs that conform to UNIX standards.

Numerics

8-name mode. See *compatibility mode*.

32-name mode. For DFSMS, it is the mode of running SMS on a DFSMS system in which up to 32 names—representing systems, system groups, or both—are supported in the SMS configuration. When running in this mode, the DFSMS system can only share SCDSs, ACDSs, and COMMDSs with other DFSMS systems running in 32-name mode.

Index

Special Characters

% freespace 122
&ACCT_JOB variable 269, 282
&ACCT_STEP variable 269, 282
&ACSENVIR variable 157, 270, 278
&ALLVOL variable 270, 281
&ANYVOL variable 271, 281
&APPLIC variable 271
&BLKSIZE variable 271
&DD variable 271
&DEF_DATACLAS variable 271
&DEF_MGMTCLAS variable 271
&DEF_STORCLAS variable 272
&DSN variable 272, 281
&DSNTYPE variable 272, 277
&DSORG variable 272
&DSOWNER variable 272
&DSTYPE variable 273
&EXPDT variable 273
&FILENUM variable 273
&GROUP variable 273
&HLQ variable 273
&JOB variable 273
&LABEL variable 274
&LIBNAME variable 274
&LLQ variable 274
&MAXSIZE 47
&MAXSIZE variable 274, 277
&MEMHLQ variable 274, 278
&MEMLLQ variable 274, 278
&MEMN variable 274, 278, 281
&MEMNQUAL variable 274, 278
&MSPARM variable 163, 275
&MSPDEST variable 163, 274
&MSPOLICY variable 163, 275
&MSPOOL variable 164, 275
&MSVGP variable 275
&NQUAL variable 275
&NVOL variable 275
&PGM variable 275
&RECORG variable 276
&RETPD variable 276
&SIZE 47
&SIZE variable 276, 277
&SYSNAME variable 278
&SYSPLEX variable 278
&UNIT variable 276, 277
&USER variable 277
&XMODE variable 277

Numerics

32-name support 12

A

ABACKUP 298
Abackup copy technique 83
ABARS
  processing
    pre-DFSMS management classes 200
ACBJBAOB JCL 347
ACBQADM2 412
ACBQFLM1 412
accessibility 95, 419
accessing
  DASD volume 57
  optical volume 63
  pool storage group 48
  remote authorization 306
  VIO storage group 48
ACDS
  allocating 13
  definition 9
  JCL allocation 13
  recovery 215
  relationship with SCDS 10
  saving 11
  share options 14
  size calculation 10
ACS
  class determination 167
  data set rename 164
  designing and testing ACS routines 137
  DFSMS NaviQuest 137
  environment
    RMMPOOL 166
    RMMVRS 166
  language
    boolean expressions 280
    comparison operators 280
    description 137
    special functions 281
    statements 284
    writing routines 265
  object
    deleting from SCDS 169
    information 168
  read-only variables 234
  read-write variables 267
  routine
    application selection, writing 139
    constants 265
    creating 138
    Distributed FileManager/MVS 101, 282
    dummy and SUBSYS data sets 138
    environments 156
    exit 163
    invoking ISPF/PDF editor 140
    language reference 265
    masks 265
    operations 137

© Copyright IBM Corp. 1986, 2002

ACS *(continued)*
routine *(continued)*
output listing 142
SCDS validation 146
storage class example 292
SYSIN and SYSOUT data sets 138
testing 148
translating 140
translation listing 142
translation results 141
validating 143
validation results 145
storage group determination 167
variables
read-only 269
read-write 267
VSAM RLS 238
ACS routine changes 333
ACS routines 127, 137
defining 137
description 4
SMS class 5
storage group 5
test cases 1
testing 1, 5
translating 1
z/OS Network File System 138
ACTIVATE, ISMF command 172
activating DFSMS configuration 338
additional volume amount 115, 206
address space
recovery 217
starting 25
Admin or User command Backup 80
aggregate group 5
altering 206
application 130
attributes
altering 207
defining 132
editing 132, 207
backup 135
backup attributes 130
copying SMS 209
defining 129, 130, 135
definition 3
displaying list 200
instruction data set 130
names 132, 207
panels 130
planning 130
recovering 135
sort criteria 200
view criteria 200
aggregate group application
understanding 129
alias device type 189
allocation
ACDS 13
COMMDS 14
control data set 9
allocation *(continued)*
data set 102
extended addressability 117
retrying 119, 120
SCDS 13
spreading across multiple volumes 106
storage class, specific volume 98
threshold 54
allocation/migration threshold 54
ALTER(YES) keyword 255
altering
aggregate group 206
data class 204
management class 202
storage
class 203
group 200
ALX parameter 47
array DASD 95
assigning
data class 127
management class 84
storage class 101
attributes
aggregate group 130, 132, 207
data class
record 113
space 112, 113
VSAM 120
GDG 78
management class
backup 79
expiration 74
migration 76
object class
backup 83
transition 81
point-in-time copy 95
pool storage group 51
retention period 75
storage class
accessibility 95
availability 88, 95
defining for CF cache sets 99, 250
definition 87
storage group 45, 46
sustained data rate 94
timing 39
VSAM 121
attributes (data class)
data class
VSAM 115
AUDIT parameter, bypass RACF logging 227
auto backup 80
storage group field 52
auto dump 52
auto migrate 51
AUTOALTER(YES) keyword 256
Automated Tape Library Dataserver (ATLDS) 42
automated testing, overview 326
automatic class selection (ACS) 4

automatic interval migration 51
availability 88, 95
availability, VSAM RLS 238
Avgrec 114

B

backup 309
  auto backup 52
  defining aggregate groups 135
  management class 79
  object copies 60
backup control data set (BCDS) 311
backup copy technique 81
backup devices 96
backup frequency 79
backup-while-open (BWO) 125
base configuration
  active control data set 3
  adding a system 32
  altering 1
  CF cache structures 247
  completing 35
  control data set application selection 34, 248
  coupling facility definition 34, 248
  defining 1, 27
  deleting a system 33
  description 3
  displaying 1
  planning 27
  renaming a system 35
  SMS complex 3, 5, 32
  testing 1
  updating 1
  VSAM RLS 33, 34, 248
base line 343
base line test set 323
BCDS 311
bias 92
block size limit 123
boolean expressions, ACS routine 280
browsing, ACS routine translation 141, 143
buffering, system-managed 117

C

cache
  device 90
  statistics 184
cache fast write function 305
cache set 34
capabilities, filtering 324
capacity planning information 309
CBROAM*xx* member 40, 70
CF cache structure
  recovering 260
CFRM (coupling facility resource manager) policy 238
changes
  ACS routine 333
  construct 333
channel-to-channel adapter device number 135
CICS 125, 235, 241, 243, 262
CIsize data 122
class transition attributes 70
classification, data 325
collecting data for input 328
collecting, data 324
collection
  assigning 43
  definition 41
  multiple 43
  name
    OAM 41
    qualifiers 275
  OAM object 64
commands
  QRETRIEV 410
  QSAVE 410
COMMDS
  allocating 14
  current utilization statistics 10
  definition 10
  JCL allocation 14
  recovery 216
  size calculation 12
  sizing 10
  SMS complex 10
compaction 118
comparing regression testing 334
comparison function 340
comparison operators, ACS routine 280
compatibility mode 28
concurrent copy 43, 85, 86, 95
connection
  storage group
    to DASD volume 57
    to optical volume 63
  system to storage group 48, 237
constants, ACS routines 265
construct changes 333
constructs
  SMS configuration 3
  types 3
control data set
  allocating 4, 9
  application selection panel 27, 34
  background 4
  description 9
  fixed data fields 10
  ISMF primary option menu 2
  multivolume 12
  RACF 233
  recovery 215
  sharing 239
  sizing 10
  structure changes 9
  types
    active control data set (ACDS) 9
    communications data set (COMMDS) 9
    source control data set (SCDS) 9
convert status for a DASD volume 39
converting to SMS management 339

converting volumes to SMS 213
CONVERTV command, DFSMSdss 156, 213, 231
COPYFILT macro 416
copying SMS classes, storage and aggregate groups 209
coupling facility (CF) 87
  cache and lock structures 238, 241
  cache structures 255
  defining cache structures 247
  displaying information 253
  lock structure 242, 256
  monitoring 253
  recovery 259
  requirements for VSAM RLS 235
  resource manager (CFRM) policy 238
  statistics monitoring 259
  system-managed duplexing 245
  system-managed lock structure duplexing 235, 242, 260
  varying cache structures 258
creating
  customized tape reports 406
  data set name reports 403
  model commands 412
  SMS reports 402
  volume reports 405
cross-system coupling facility (XCF) 5
cycle end time 59
cycle start time 58

D

DASD 194
  fast write 90
  fast write function 305
  storage
    control unit 303
    groups 38, 57
    object 39
  volume
    access by storage group 57
    data set space 276
    deleting from the system 210
    status 38, 39
DASD volume 37
  initial status 39
DASDVOL alter authority 307
DASDVOL authority 57, 102
data class 127
  altering 1, 204
  assigning 127
  attributes
    description 112
    JCL processing 127
    Recorg 127
  creating additional 128
  defining 1, 111
    application selection 112
    record and space attributes 113
    volume 115, 120
  definition 111
data class *(continued)*
  deleting 212
  description 3
  displaying 1
  list 195
  order when processing attributes 127
  planning 111
  record and space attributes 113
  specifying sort criteria 195
  understanding 111
  view 195
  volume attributes 115
  VSAM attributes 115, 120
data classification 323, 325
data collection 309
data DCOLLECT, input 324
data saved lists, input 324
data set 125
  allocating space 31
  allocation 102, 277
  duplicate names 228
  empty 112
  extended addressability 117
  instruction 130
  mask 265
  name qualifiers 275
  name type 116
  nonrecoverable 238
  null 112
  open for VSAM RLS 253
  organization
    DSORG value 121
  qualifiers 281
  read-only variables on rename 278
  recall processing 165
  RECORG 276
  recover processing 165
  recoverable 238
  renaming 278
  selection 130
  separation 31, 315
data set access
  initial access response seconds 94
data set name reports, creating 403
data set placement
  primary list 103
  secondary list 103
  tertiary list 103
data set separation 103
data set stacking
  description 160
  read-write variables 268
  using
    VOL=REF 160
    VOL=SER= 160
data SMF, input 325
data subtype 323
data testing, example 339
data type 323
data VMA, input 325

data-in-virtual (DIV)
  size limit 12
data, collecting 324
DB2
  object storage table 39
DCOLLECT command
  active
    data set 310
    volumes 310
  backup data 310
  capacity planning data 310
  description 309
  ISMF panels 309
  migration data 310
  pool storage groups 312
  storage groups 312
  volumes 312
default device geometry parameter 31
default management class
  SCDS base define field 30
  specifying 69, 147
default management class parameter 30
default unit parameter 31
DEFINE CLUSTER command 97
defining CF cache and lock structures 241
deleting
  DASD volumes from the system 210
  SMS classes 212
  storage groups 211
destaging 306
determining errors 336
device information 303
device status 188
DEVSERV command 188
  NOT-NPAV 193
  PATHS 194
  QDASD 194
  QPAVS 189, 190, 192
    NOT-ALIAS 193
    NOT-NPAV 189
  sample output 190
  UNBOX 192
DFSMS 1
DFSMS configuration, activating 338
DFSMSrmm 1
  scratch tape pools 42
  storage groups 42
  tape storage groups 42
  Vital Record Specification (VRS) 71
diagram, tree 325
directory 114
disability 419
DISALL
  storage group to optical volume 63
  system to storage group 48
disaster recovery 95
DISNEW
  storage group to optical volume 63
  system to storage group 48
DISPLAY SMS command
  cache statistics 184
DISPLAY SMS command *(continued)*
  coupling facility (CF) information 253
  OAM status 179
  SMS maintenance 188, 189
  trace information 180
DISPLAY XCF command 253
Distributed FileManager/MVS
  ACS routines
    determining SMS classes 101, 138
  using ACS routine variables 282
DIV 12
DO statement 287
drive (optical)
  displaying list 200
  sort criteria 200
  view criteria 200
drive start threshold 59, 62
DS separation profile parameter 29, 31
DSTACK, SMS data set stacking service 18, 174
dual copy 43, 85, 95
dump
  auto dump 52
  classes 52
  dump system name 52
dump class 54
duplexing 305
duplicate data set names 228
dynamic data set
  cache management 90
dynamic volume count parameter 118

E

edit a data set 133, 208
ENABLE
  storage group to optical volume 63
  system to storage group 48
END statement
  ACS routine
    description 292
end user
  ISMF primary option menu 2
enhancements, miscellaneous 410
Enterprise Storage Server (ESS) 95, 99
  FlashCopy 85
  SnapShot 85
errors
  definition 323
  determining 336
examples
  data testing 339
exceptions 340
EXECs
  ACBQADM2 412
  ACBQFLM1 412
EXIT statement 290
exit, pre-ACS routine 163
Expdt (data class field) 115
expected results 323
expected results, saving 337

expiration
  date override 19
  EXPDT 74
  management class attributes 74
  object class
    backup attributes 83
    transition attributes 81
  retention limit 74
extend storage group 53
extended addressability 117
extended format 121, 123
extended remote copy (XRC) 85

F

FALLBACK command 263
fallback, VSAM RLS 261
false lock contention 244
FIELD resource class 233
FILTLIST statement 285
FILTLISTs, updating 333
fixed data fields, sizing 10
FlashCopy 85, 86, 95
flowchart, testing example 340
formula
  CF lock structure 242, 246
  SHCDS size 240
FRlog parameter 126

G

GDG
  management attributes 78
  storage management 71
  using data set separation with 318
generating model commands 410
generating test cases 328
global connectivity, examples 237
global shared resources (GRS) 5
guaranteed backup frequency 55
guaranteed space 94, 97, 98, 99
guaranteed synchronous write 99

H

HCD (hardware configuration definition) 48
hiperspace size 19

I

IART
  value 105
IART (initial access response time) 93
ICKDSF (Device Support Facilities) 38
IF statement 288
IGDSMSxx
  creating 14
  description 14
  modifying for VSAM RLS 247
IGDSMSxx parmlib member
  CF_TIME parameter 247, 257
  DEADLOCK_DETECTION parameter 247, 257
  RLS_MAX_POOL_SIZE parameter 242, 247, 258
  RLS_MaxCfFeatureLevel parameter 247, 257
  RLS_MAXCfFeatureLevel parameter 21, 125
  RLSINIT parameter 247, 257
  SMF_TIME parameter 247, 258
IGDSSIIN
  optional keyword 24
  SMS initialization 171
IGWLOCK00, CF lock structure 242
implementation phase 323
initial access response seconds 93
initial program load (IPL) 171
initial test 323
input data
  DCOLLECT 324
  saved lists test 324
  SMF 325
  VMA 325
installation defaults 27
instantaneous replication 95
instruction data set 134
  aggregate group field 208
  definition 130
Interactive Storage Management Facility (ISMF) 1
Interactive System Productivity Facility (ISPF)
  authorizing access 233
  PDF editor 140
interval migration, automatic 51
IODF 189
IPL 171
ISMF 1, 220, 303
  ACTIVATE command 172
  aggregate group application 129
  application module names 221
  applications 1
  automatic class selection application 137
  command 295
    summary 295
  control data set 27
  data class application 111
  description 1
  dialog 13
  displaying CF information 254
  end user 2
  function module names 221
  line operator 222
  line operator module names 222
  management class application 69
  module names
    line operators 222
    locating 221
  primary option menu 2, 26, 321
  RACF interactions 220
  storage administrator 2
  storage administrator functions 1
  storage group application 37, 45
  storage group status 184

ISMF *(continued)*
variables
retrieving 410
saving 410
ISMF primary option menu
access 4
aggregate group 2
automatic class selection (ACS) 2
control data set 2
data class 2
data collection 2
library management 2
management class 2
storage administrators 4
storage class 2
storage group 2
IXCL1DSU Couple Data Set format utility 243

J

JCL keyword
DISP=OLD 85
STORCLAS=DUPT@SMS 85
UNIT= 85
VOL=SER= 85
JES3 33

K

key ranges 97
keyboard 419
Keylen 122
Keyoff 122
keywords
optional 16
required 16

L

level 0 storage 76
LIBPORT-ID field
HCD panel 42
library
displaying list 200
names, defining 59, 61
sort criteria 200
view criteria 200
library names 59, 61
LIBRARY-ID field
HCD panel 42
library, test bed 324
line operators
summary 298
LIST command 195
listing
aggregate group 200
data class 195
management class 200
optical
drive 200
library 200
listing *(continued)*
storage
class 200
group 200
tape data class 200
volumes in a storage group 66
LISTSYS line operator 184
LISTVOL line operator
listing volumes
DASD 66
mountable optical volume list 66
object and object backup 66
OAM status 179
storage group
DASD 39
tape 42
lock structure
sharing control 239
lock structure, CF
adjusting size 246
defining 242
false contention 244
retained locks 243
LOG parameter
data class 126
DEFINE CLUSTER command 253
logical subsystem 95
logstream ID 127
LRECL 114

M

macro, COPYFILT 416
maintaining 179
management class 84
ACS routine 27
altering 1, 202
application selection 71, 74
assigning 84
attributes
backup 83
defining 71, 72
expiration 74
GDG 78
migration 76
retention limit 75
backup
attributes 79
concurrent copy 81
copy technique 81
frequency 79
version retention period 80
default 27, 30, 69
defining 1, 69
defining base configuration information 27
deleting 212
describing 70
description 3
displaying 1
displaying list 200
expiration attributes 73

management class *(continued)*
NOACTION 27
OAM 70
planning 71
RACF 233
retention limit 74
sort criteria 200
understanding 69
view criteria 200
management class name 132, 207
manual tape library (MTL) 42
masks, ACS language 265
MAXSYSTEM value 243
media interchange 122
media type 121, 123
member name 133, 208
menu, ISMF primary option 321
method considerations 325
method of testing, NaviQuest 339
MGMTCLAS syntax
JCL, TSO/E, IDCAMS parameter 84
OSREQ STORE or CHANGE parameter 84
migrated storage 76
migration
auto migrate 51
automatic interval 51
data
collection 310
information 309
level 1 storage 76
level 2 storage 76
management class attributes 76
pool storage group 54
threshold 54
migration attributes 77
migration control data set (MCDS) 311
millisecond response 88
millisecond response for storage class 91
millisecond response time (MSR)
defining 89
miscellaneous enhancements 410
model commands, creating 412
modifying IGDSMSxx 247
module name, ISMF
commands 222
functions 221
line operators 222
monitoring
coupling facility (CF) 253
statistics 259
monitoring for false lock contention 245
MS volume selection 104
MSS group name 275
multiple allegiance 99
MVS
command for displaying CF information 253
MXIG parameter 47

N

naming
SHCDS 239
NaviQuest
designing and testing ACS routines 137
maintaining SMS configuration 179
storage management tasks in batch 1
NaviQuest overview 321
NaviQuest primary option menu 322
NaviQuest, how to use to test 327
NaviQuest, method of testing 339
NaviQuest, testing procedure 327
new terminology 322
NOTCON
storage group to optical volume 63
system to storage group 48
null results 343
nulls 340
number of backup versions 80
number of copies 132, 207
numeric read-only ACS variables 280

O

OAM
activating SCDSs 178
backup storage group 39
cataloging 41
class transition attributes 70
collection names 41
DISPLAY SMS command 179
management class definition 70
object collection 64
OSMC 39
restarting 178
status 179, 183
storage group
hierarchy 39
object collection 44
OAMXCF parameter, DISPLAY SMS command 183
object
access method 179
ACS information
deleting 169
displaying 168
backup storage group 39, 46, 62
cataloging 41
character strings 265
class
ACS routines 265
backup attributes 83
transition attributes 81
collection 41, 43
DB2 directory 41
mask representation 265
name 274
reading 40
status field 62
storage group
access types 63

object *(continued)*
storage group *(continued)*
ACS routines 265
attributes 58, 60
backup 6, 46
defining 57
definition 39
devices 63
hierarchy 39, 46
qualifier 39
SCDS 66
timing attributes 39
storage hierarchy 86
write requests 62
writing 40
object access
initial access response seconds 93
optical
library
names 59, 61
pseudo 40
sharing 40
volume
access 63
label 40
table of contents 40
option 11, primary option menu 321
OSMC
storage management cycle 70
OSMC Processing System 59
OSREQ STORE macro 74
output data set name prefix 132
overflow storage group 53
overview automated testing 326
overview, NaviQuest 321

P

parallel access volumes (PAV) 99
Parallel Sysplex
cross-system coupling facility (XCF) 5
definition 6
displaying OAM status 183
multiple SMS complexes 5
MVS environment 5, 32
VSAM RLS 238
partial release 77, 206
password protection 219
PAV
PAV-alias 189
PAV-base 189
PDSE
monitor 20
sharing 5, 20
peer-to-peer remote copy (PPRC) 85
phase test set 323
physical sequential data set 94
planning
aggregate groups 130
base configuration 27
data class 111
planning *(continued)*
storage
class 87
group 43
point-in-time copy
concurrent copy 95
FlashCopy 95
SnapShot 95
virtual concurrent copy 95
PPRC 85
pre-ACS interface
DFSMSrmm support 156
pre-ACS routine exit 163
prefix, subtype 324
primary
location 130
storage 76
primary list 102, 104
volume 105
primary option menu, ISMF 321
primary option menu, NaviQuest 322
PROC statement 285
program control feature 220
pseudo optical library 40

Q

QDASD parameter 189
QRETRIEV command 410
QSAVE command 410
qualifier field 58
QUIALL, SMS status 48
QUINEW, SMS status 48

R

RACF 219, 220
authorization structure 221
bypass logging 227
DASDVOL alter authority 307
FACILITY profiles for storage administration 227
FIELD resource class 233
ISMF functions 220
program control feature 220
protecting modules 227
RDEFINE command 227
resource owner 220
RACF security 5
RAMAC 93
RAMAC Virtual Array 12
RAUTH line operator 306
RDEFINE command 227
read caching 304
read-only variables
ACS routine 280
initializing 277
OAM 278
pre-ACS routine exit 163
restrictions 279
reason code 190
recall processing 165

Recfm record format 114
recommendations
  defining pool storage groups 38
  defining VSAM RLS attributes 252
  determining hardware requirements for VSAM RLS 235
  disabling SMSVSAM 262
  dynamic data set cache management 92
  information recovery 215
  OAM dependency on INTERVAL keyword 175
  protecting ISMF functions 220
  specifying LRUCYCLES value 19
  specifying LRUTIME value 20
  STGADMIN.DPDSRN 228
  using cache/DFW 92
  using CF cache structures in JES3 environment 33
  using control data sets 215
  using data set separation 315, 317
  using NaviQuest 328, 349
  using Naviquest test cases 328, 329, 331, 333, 380
  using overflow storage groups 53
  using sharing control data sets 239
  using the INTERVAL keyword 19
  using the RLS_MAX_POOL_SIZE parameter 242
  using VOL=REF vs. VOL=SER 160
  writing the SMS record 24
record, data class attributes 113
recording technology 123
Recorg 121
recover processing 165
recovering
  ACDS 215
  COMMDS 215
  control data sets 215
  SCDS 215
  SMS address space 215
recovery
  location 130
  SMS
    address space 217
    complex 217
    information 215, 217
  VSAM RLS 259
reducing false lock contention 245
regression test 323
regression testing, comparing 334
remote access 306
rename data set, name in use 228
reports, creating 405
requirements
  changing SMS configurations 178
  creating a data set separation profile 317
  defining shareoptions for data class 124
  defining storage classes for VSAM RLS in a JES3 environment 250
  defining the base configuration 28, 32
  defining the CF lock structure 242
  determining hardware requirements for system-managed lock structure duplexing 236
  determining hardware requirements for VSAM RLS 235
requirements *(continued)*
  displaying storage group status using ISMF 184, 185, 186
  enabling extend processing 38
  enabling VSAM RLS processing 252
  manually activating the first SMS configuration 171
  restarting the SMSVSAM server 261
  specifying an extend storage group 54
  using ACS routines 138
  using NaviQuest 326, 327, 332, 344
  using read-write variables in ACS routines 267
  using tailored compression 17
  using the COPY BYPASSACS command 154
  using the RESTORE BYPASSACS command 155
resource class, FIELD 233
resource owner 219
restarting OAM 178
restrictions
  allocating control data sets 9, 10, 12, 21
  assigning DASD storage groups to data sets 57
  defining data class attributes 116, 119
  defining object class transistion attributes 82
  defining the base configuration 28
  information recovery 215
  managing VSAM data sets allocated using IDCAMS 293
  modifying caching characteristics 304
  modifying destaging characteristics 306
  processing data class attributes in JCL 128
  reinitializing a storage subsystem 306
  running SMS in a Parallel Sysplex environment 5
  SMS volume selection for data set allocation 102
  specifying attributes for BWO and recovery 125
  testing ACS routines 148
  using ACS routines 138
  using aggregate groups 129
  using control data sets 215
  using Naviquest test cases 380
  using OAM read-only variables 278
  using read-only variables 279
  using SMF data with NaviQuest 325
  using system-managed lock structure duplexing 260
  using the CF for VSAM record-level sharing 100
  using the FILTLIST VALID_UNITS INCLUDE statement 293
  using the RLS_MaxCfFeatureLevel parameter 175, 258
  using the SETSMS operator command 173, 178, 200
  using VMA data with Naviquest 325
results, expected 323
retain days extra backup versions 80
retain days only backup version 80
retained locks 243
retention limit 74
retention period
  limit 74
  management class comparison 75
  read-only variable 276
Retpd (data class field) 115

retrieving ISMF variables 410
REUSE 13
RLS CF cache value parameter 125
RLS_MAXCfFeatureLevel
  IGDSMSxx member 21, 247
  SETSMS command 175
RLSINIT parameter 262
rolled-off GDS 78
running test cases 333

S

saving
  expected results 337
  ISMF variables 410
SCDS 7, 111
  ACS routine validation 146
  allocation 13
  backup 10
  base configuration
    installation defaults 27
  base configuration definition 27, 34
  changing SMS configurations 177
  copying object storage group 66
  coupling facility definition 34, 248
  creating 10
  defining status 28
  definition 9
  error 12
  JCL allocation 13
  modifying 10
  multivolume 12
  recovery 215
  relationship with ACDS 10
  saving 11
  size calculation 10
  SMS configuration 9
  system group names 27
  system names 27
  updating 10, 209
SCDS Status field 30
scenario, testing 339
SDR
  value 94, 95
secondary list 102
SELECT statement 289
selection data set 133
  aggregate group field 208
  description 130
sequential data striping
  batch processing 94
  sustained data rate 94
SET operator command (T SMS command) 25, 171
SET statement 286
SETCACHE function 303
SETCACHE line operator 303
SETOAM statement 40
SETOSMC statement 70
SETSMS operator command 12, 25, 172, 173
setup, testing environment 326
SETXCF START command 245, 255, 256
Shareoptions parameter 124
SHAREOPTIONS parameter 14
  specified for an SHCDS 239
sharing control data set (SHCDS) 239
SHCDS 230, 238, 239, 254
  recovering 261
  share options 239
SHCDS (sharing control data set) 239
SHCDS command 247
shortcut keys 419
size
  CF cache structure 241
  CF lock structure 242
  SHCDS 240
SMF records
  CF-related 258
  interval time 16, 173
  test case creation 332
SMS 42, 179
  32-name support 12
  activation 9
  active
    configuration information 179
    data set information 309
    volume information 309
  address space 25, 217
    canceling 217
  aggregate group 1
  base configuration 1, 3
  cache statistics 184
  capacity planning information 309
  classes, determining 167
  complex 3, 13, 32
  complex recovery 217
  configuration 4, 9
    activating 171
    automatic activation 172
    changing parameters 173
    content 3
    manual activation 171
    minimal 146
  converting volumes to SMS 213
  copying classes, storage groups, aggregate groups 209
  defining SMS through IEFSSNxx 23
  defining to z/OS 23
  definition 1
  deleting classes 212
  device status 188
  IGDSMSxx 14, 15
  information recovery 215
  initialization 9
  initialization parameters 14
  initializing
    IGDSMSxx Member 14
  IPL of SMS complex systems 171
  management class, default 30
  management policy 3, 4
  operator commands
    DISPLAY SMS command 179
    VARY SMS command 179

SMS *(continued)*
optional keywords 16, 24
Parallel Sysplex 5, 32
Parallel Sysplex environment 1
physical storage 37
pre-ACS interface 156
preliminary steps 9
preparing 4
protecting 219
RACF
function 220
STGADMIN profiles 227
record definition 24
recovering
ACDS 215
address space 215
COMMDS 215
control data sets 215
SCDS 215
required keywords 16
resource owner 220
SMS class 1
storage group 1
deleting 210
determining 167
LISTSYS line operator 184
moving volumes 201
redefining volumes 201
storage group status 60, 62
storage management policy 1
SYS1.PARMLIB 15
system
group 5, 32
name 5, 32
system-managed storage 1
TRACE information 180
volume status 187
writing the SMS entry 23
SMS class 5
naming 3
SMS complex
definition 6
system group 5
SMS configuration
ACS routine 3
activating 231
base configuration 9
class information 3
drive definition 3
FIELD resource class 231
group information 3
optical library definition 3
RACF authority 231
tape library definition 3
validating 5
SMS management, converting to 339
SMS reports, creating 402
SMS volume selection
primary list 105
randomizing technique 105
secondary list 105
SMS volume selection *(continued)*
space constraint relief 105
tertiary list 105
SMSVSAM address space 252
recovering 261
SnapShot 85, 86, 95
sort criteria
aggregate group 200
data class 195
management class 200
optical
drive 200
library 200
storage
class 200
group 200
source control data set (SCDS) 3
SMS configuration 9
source volume 95
space 114
attributes 113
constraints 119, 120
use information 309
space constraint relief 119
statistics
active configuration 179
cache 184
coupling facility monitoring 259
device 188
OAM 179
storage group
LISTSYS line operator 184
volume 187
STGADMIN profiles
command and keyword related 227
Facility Class Profiles 231
storage administrator
authorizing access 233
control data set application selection 27, 34, 248
ISMF primary option menu 2
primary option menu
accessing 26
storage class 99
accessibility attributes 95
ACS routine sample 292
altering 1, 203
application selection panel 87, 99, 250
assigning 101
attributes 87, 99, 250
sequential data striping 85
VSAM RLS 85
availability attributes 88, 95
bias 92
continuous availability 43
copying SMS 209
coupling facility 87
DASD data sets
system-managed 85
data set placement 103
defining 1, 87
defining additional 101

storage class *(continued)*
definition 250
deleting 212
description 3, 85
device failure 95
displaying 1
displaying list 200
dual copy 95
guaranteed space 97
initial access response seconds 93
millisecond response 88, 91, 92
objects 86
performance 88
planning 87
point-in-time copy attributes 95
protection 233
RACF 233
sort criteria 200
tape data sets 86
tape volumes 86
understanding 85
view criteria 200
volume allocation 98
VSAM RLS 99, 251
storage control unit 303
storage group 1, 5
ACS routine 98, 167
additional groups 66
altering 200
application selection panel 45
assigning 64
attributes 45, 46
auto migrate 51
automatic interval migration 51
backup object attributes 60
changing 200
convert status 39
copying SMS 209
DASD volume
SMS complex 56
status 38
defining 37, 45, 51
defining volume status 56
deleting 210
description 3
displaying list 200
dummy 37
attributes 45, 49
defining 49
volume serial numbers 50
dump classes 54
eligible volumes 94
extend 53
LIST commands 195
location 39
migration threshold 54
multiple 40
name 45
naming 3
OAM collection names 41
object 37, 57
storage group *(continued)*
attributes 58
backup 46
directory 44
storage hierarchy 39
object backup 37, 46, 60
optical volumes 63
overflow 53
planning 43
planning for OAM 43, 44
pool 37, 45, 46, 51, 53
primary space 94
qualifier 39
sort criteria 200
status
LISTSYS line operator 184
volume 56
status field 55
Status field 65
storage panels 46
system access 48
tape 37, 46
defining 64
library definition 42
SMS support 42
type 37, 45
types 39
view criteria 200
VIO 37, 45
defining 46
volume
characteristics 38
moving 201
redefining 201
Storage Management Subsystem (SMS) 1
striping volume selection 106
subtype
prefix 324
test set 324
testing next 338
sustained data rate (SDR) 94
SYS1.PARMLIB 32
creating 4
IEASYSyy member 14, 23
IEFSSNxx member 14, 23
IGDSMSxx member 14, 247
initializing 14
modifying 4, 9, 14, 15
SMS 4
SMS complex 4
system group
common configuration 5
definition 6
name 6
system name 52
definition 6
system temporary data sets 340
system-managed
data sets 3
objects 3
systemp 340

T

T SMS command 25, 171
tape library
  SMS support 42
tape management system support 163
tape mount management 52, 54, 325
target volume 95
tasks
  using data set separation
    specifying the data set separation profile 31
terminology, new 322
tertiary list 102
test bed library 324
test case, definition 324
test cases
  generating 328
  running 333
test data restrictions 325
test results, validating 336
test set, subtype 324
testbed library attributes 327
testing
  ACS routine 148
  ACS routines 326
  environment setup 326
  initial 323
  next subtype 338
  phase set 323
  running ACS routine cases 151
  scenario 339
track images 95
transactional VSAM 15
translating ACS routines 333
translation
  ACS routine
    description 140
    running with mixed levels of DFSMS 147
tree diagram 325
TSO (time sharing option), authorizing access 233

U

UCB 189
UNBOX 189
unit affinity 161
unlisted device 190
updating
  FILTLISTs 333

V

validating
  ACS routines 333
  test results 336
validation
  ACS routine 147
    conditions 144
    entire SCDS 146
  browsing ACS routine results 145
  individual ACS routine 143
variables, read-only
  ACS routine 269
  definition 269
  initializing 277
  list 269
  OAM 278
  restrictions 279
variables, read-write
  definition 267
  list 267
VARY SMS command 258
versioning devices 96
VIEW CRITERIA (data class field) 195
VIO
  Maxsize 47
  storage group allocation 45
  unit 48
virtual concurrent copy 86, 95
Vital Record Specification (VRS) 71
VMA (volume mount analyzer) 325, 327, 331
  Naviquest test case 383
volume
  data class 115
  DISPLAY SMS command 187
  dummy storage group 50
  selection 102
    enhanced 103
    rules 107
  serial number comparison 282
  status 187
  storage characteristics 38
  VSAM attributes 115
VOLUME 189
volume count 115
  dynamic 118
  how SMS determines 119
volume full at write error 60, 62
volume full threshold 59, 61
volume serial label 194
VSAM
  attributes (data class) 120
  data class attributes 120
  data set 94
  linear data set
    SMS.SCDS1.SCDS 13, 14
  Record-Level Sharing (RLS) 33
  storage groups 38
VSAM record-level sharing (RLS) 5
VSAM RLS
  activating 252
  adjusting lock structure size 246
  administering 235
  authorization 247
  authorizing access 247
  base configuration 33
  CF cache and lock structures 241
  coupling facility (CF) 99
  displaying CF information 253
  environment 236
  fallback processing 261
  false lock contention 244

VSAM RLS *(continued)*
hardware requirements
coupling facility (CF) 235
lock structure 242
monitoring the coupling facility (CF) 253
planning for availability 238
preparing 235
recovery 259
requirements 252
retained locks 243
SHCDS
defining 239
storage class definition 250
SYS1.PARMLIB IGDSMSxx member
modifying 247
system connectivity requirements 237
VTS (Virtual Tape Server) 275
initial access response seconds 93

W

WRITE statement 291
writing SMS entry 24

X

XCF 5
XES coupling definition process 242
XRC 85

Readers' Comments — We'd Like to Hear from You

z/OS
DFSMSdfp Storage Administration Reference

Publication No. SC26-7402-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC26-7402-01

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
RCF Processing Department
M86/050
5600 Cottle Road
San Jose, CA U.S.A. 95193-0001

Fold and Tape | **Please do not staple** | Fold and Tape

SC26-7402-01

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-7402-01