

z/OS  
3.2

*DFSMSdfp Storage Administration*



**Note**

Before using this information and the product it supports, read the information in [“Notices” on page 397](#).

This edition applies to IBM® z/OS® 3.2 (5655-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2025-09-30

© **Copyright International Business Machines Corporation 1984, 2025.**

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

---

# Contents

<b>Figures.....</b>	<b>xiii</b>
<b>Tables.....</b>	<b>xvii</b>
<b>About this document.....</b>	<b>xix</b>
Required product knowledge.....	xix
Notational conventions.....	xix
<b>z/OS information.....</b>	<b>xxi</b>
<b>How to provide feedback to IBM.....</b>	<b>xxiii</b>
<b>Summary of changes.....</b>	<b>xxv</b>
Summary of changes for z/OS 3.2.....	xxv
Summary of changes for z/OS 3.1.....	xxv
<b>Chapter 1. Introducing the Storage Management Subsystem.....</b>	<b>1</b>
Understanding the Storage Management Subsystem.....	1
Using the Interactive Storage Management Facility.....	1
Customizing ISMF.....	2
Defining your storage management policy.....	3
Preparing for and implementing the Storage Management Subsystem.....	5
Running SMS in a parallel sysplex environment.....	6
Basic terms and definitions.....	6
System grouping.....	6
Running SMS on mixed DFSMS releases.....	9
<b>Chapter 2. Preparing for the Storage Management Subsystem.....</b>	<b>11</b>
Allocating control data sets.....	11
Source control data set (SCDS).....	11
Active control data set (ACDS).....	12
Communications data set (COMMDS).....	12
Calculating the size of storage and active control data sets.....	13
Calculating the size of a COMMDS.....	14
Selecting volumes for control data sets.....	15
Allocating an SCDS.....	15
Allocating an ACDS.....	15
Allocating a COMMDS.....	16
Recommendations.....	16
Defining SMS as trusted to RACF.....	17
Modifying the SYS1.PARMLIB data set.....	17
Initializing SMS through the IGDSMSxx member.....	17
Starting the SMS address space.....	17
Accessing the storage administrator primary option menu.....	18
Planning for VSAM record-level sharing.....	18
<b>Chapter 3. Creating the base configuration.....</b>	<b>19</b>
Planning the base configuration.....	19

Defining the base configuration.....	19
Validating the SCDS.....	20
Activating the SCDS.....	20
Specifying the default management class.....	21
Specifying the default unit.....	21
Specifying the default device geometry.....	21
Specifying the DS separation profile.....	21
Specifying systems and system groups in the SMS complex.....	22
Defining the base configuration for VSAM record-level sharing.....	23
Using cache structures.....	23
Defining cache sets.....	23
Defining lock sets.....	24
Completing the base configuration.....	24
<b>Chapter 4. Defining storage groups.....</b>	<b>25</b>
Understanding storage groups.....	25
DASD storage groups.....	25
DASD volume status for data sets.....	26
Object and object backup storage groups.....	27
OAM collection names.....	29
Tape storage groups.....	29
Planning storage groups for data sets.....	30
Planning storage groups for OAM object collections.....	31
Defining storage group attributes.....	32
Storage group types.....	33
Defining a VIO storage group.....	34
Defining a pool storage group.....	35
Assigning DASD storage groups to data sets.....	43
Defining a dummy storage group.....	44
Defining a copy pool backup storage group.....	45
Defining an object storage group.....	46
Defining an object backup storage group.....	48
Defining object or object backup storage group status.....	50
Assigning an OAM object collection to a storage group.....	51
Defining a tape storage group.....	51
Defining tape storage group status.....	52
Defining additional storage groups.....	52
Listing volumes in a storage group.....	53
<b>Chapter 5. Defining reserve storage pools.....</b>	<b>55</b>
Understanding reserve storage pools.....	55
Assigning volumes to a reserve storage pool.....	55
<b>Chapter 6. Defining cloud network connections.....</b>	<b>57</b>
Understanding clouds.....	57
Planning a cloud .....	57
Defining a cloud.....	58
Steps for defining a DS8000 as an object proxy.....	58
Setting up digital certificates for the cloud.....	59
Setting up mutual certification.....	59
Setting up digital certificates when using DS8000 as an object proxy.....	60
Defining additional clouds.....	60
<b>Chapter 7. Defining network connections.....</b>	<b>61</b>
Understanding network connections.....	61
Securing your network communication.....	61
Setting up digital certificates for the cloud.....	61

Setting up certificates signed by a certificate authority.....	61
Setting up self-signed certificates.....	62
Object storage network connections.....	63
Planning network connections.....	63
Defining network connections attributes.....	63
Defining connections to object storage clouds.....	63
Defining connections to a TS7700 for DS8000 object store.....	63
Defining connections to an s3-based object storage cloud service provider.....	64
Defining connections to an OpenStack Swift-based object storage cloud service provider.....	65
Defining additional networks.....	66
<b>Chapter 8. Defining management classes.....</b>	<b>67</b>
Understanding management classes.....	67
Default management class.....	67
OAM management classes.....	67
Describing management classes.....	68
Planning management classes.....	68
Defining management class attributes.....	69
Defining management class expiration attributes.....	70
Defining management class migration attributes.....	73
Defining management class backup attributes.....	77
Defining class transition attributes.....	79
Defining aggregate backup attributes.....	80
Assigning management classes.....	81
Defining additional management classes.....	82
<b>Chapter 9. Defining storage classes.....</b>	<b>83</b>
Understanding storage classes.....	83
Storage classes for data sets.....	83
Storage classes for objects.....	84
Planning storage classes.....	84
Defining storage class attributes.....	84
Defining performance objectives.....	85
Defining availability.....	92
Defining accessibility.....	93
Defining the guaranteed space attribute.....	95
Defining guaranteed synchronous write.....	98
Defining use of the coupling facility for VSAM record-level sharing.....	98
Defining when to disconnect the sphere.....	99
Defining use of zHyperLinks.....	99
Assigning storage classes.....	100
Defining additional storage classes.....	101
SMS volume selection for data set allocation.....	101
Volume selection preference attributes.....	101
Conventional volume selection.....	102
Striping volume selection.....	105
Tuning considerations.....	108
Cluster and storage facility image considerations.....	108
Spreading allocations across multiple volumes.....	109
Space constraint relief.....	109
Using the multi-tiered storage group function.....	109
Using the parallel access volume option.....	109
Possible reasons for volume selection failure.....	113
Defining secondary lock tables.....	114
<b>Chapter 10. Defining data classes.....</b>	<b>117</b>
Understanding data classes.....	117

Planning data classes.....	117
Defining data class attributes.....	118
Defining record and space attributes for data class.....	118
Defining volume and data set attributes for data class.....	121
Specifying attributes to handle space constraints during allocation.....	126
Defining VSAM attributes and specifying media types for data class.....	128
Defining the encryption management mechanism.....	131
Defining Shareoptions and RLS attributes for data class.....	131
Specifying attributes for data set reuse and loading.....	133
Specifying attributes for backup-while-open (BWO) and recovery.....	133
Assigning data classes.....	135
Specifying data classes outside ACS routines.....	135
Processing data class attributes in JCL.....	135
Defining additional data classes.....	136
<b>Chapter 11. Defining aggregate groups.....</b>	<b>137</b>
Understanding aggregate groups.....	137
Planning aggregate groups.....	138
Defining aggregate groups.....	138
Defining aggregate group attributes.....	139
Editing aggregate group attributes.....	139
Backing up and recovering an aggregate group.....	140
Defining additional aggregate groups.....	141
<b>Chapter 12. Defining copy pools.....</b>	<b>143</b>
Planning a copy pool.....	143
Defining a copy pool.....	144
Steps for defining a copy pool.....	144
Defining a copy pool backup storage group.....	148
<b>Chapter 13. Defining ACS routines.....</b>	<b>149</b>
Understanding ACS routines.....	149
Using ACS routines for data sets created by z/OS Network File System and Distributed File Manager/MVS.....	150
Restrictions on using ACS routines.....	150
Creating ACS routines.....	150
Translating ACS routines.....	151
Browsing the results of a translation.....	151
Browsing the results of an unsuccessful translation.....	152
Validating ACS routines or an entire SCDS.....	153
Validating an ACS routine.....	153
Validating an entire SCDS.....	155
Translating and validating in a sysplex environment.....	156
Testing ACS routines.....	157
ACS test usage.....	157
Creating ACS test cases.....	157
Running ACS test cases.....	158
ACS routines invoked for copying and importing data sets.....	158
ACS routines invoked for restoring, recalling, recovering, and converting data sets.....	160
ACS routine environments.....	162
JCL DD statement (batch), and dynamic allocation.....	162
Volume reference.....	162
Data set stacking.....	164
Non-data set stacking allocations.....	166
Tape management system support.....	168
Access Method Services.....	169
DFSMSHsm.....	170

DFSMSdss.....	170
DFSMSrmm.....	170
ISMF.....	171
OAM.....	171
Processing of SMS classes and storage groups.....	172
Displaying ACS object information.....	172
Deleting an ACS object from an SCDS.....	173
Using security labels in ACS routines .....	173
Planning and installing .....	174
Administering.....	175
<b>Chapter 14. Activating Storage Management Subsystem configurations.....</b>	<b>179</b>
Manually activating the first Storage Management Subsystem configuration.....	179
Step one: IPL each system in the SMS complex.....	179
Step two: prepare one system.....	179
Step three: activate the configuration from one system.....	179
Step four: activate SMS on the other systems.....	180
Automatically activating a Storage Management Subsystem configuration.....	180
Converting the SMS configuration from compatibility to 32-system mode.....	180
Displaying storage management subsystem information.....	181
Controlling DFSMStvs processing.....	181
Monitoring application programs that use DFSMStvs.....	182
Changing DFSMStvs status.....	182
Changing Storage Management Subsystem parameters.....	182
Parameters of the SETSMS operator command.....	182
SETSMS command to alter the setting of BreakPointValue.....	183
SETSMS command to alter the setting of USEEAV.....	183
Considerations when changing Storage Management Subsystem configurations.....	183
OAM considerations when changing SCDSs.....	184
<b>Chapter 15. Maintaining the Storage Management Subsystem.....</b>	<b>185</b>
Displaying SMS and OAM information.....	185
Displaying information about the active configuration.....	185
Displaying SMS TRACE information.....	186
Displaying storage group status using the DISPLAY SMS command.....	187
Displaying OAM status in a parallel sysplex.....	189
Displaying the caching statistics.....	189
Displaying storage group status using ISMF.....	189
Displaying volume status using ISMF.....	190
Displaying volumes using the DISPLAY SMS command.....	192
Displaying the device status.....	192
Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF.....	200
Listing with view and sort.....	200
Listing data classes.....	201
Listing storage groups, management classes, storage classes, aggregate groups, libraries and drives.....	202
Altering data set associations.....	202
Altering SMS components.....	202
Altering storage groups.....	203
Redefining volumes among storage groups.....	203
Altering management classes.....	204
Altering storage classes.....	205
Altering data classes.....	205
Altering copy pools.....	206
Altering aggregate groups.....	207
Altering the SCDS on different DFSMS releases.....	208
Copying SMS components.....	208

Deleting DASD volumes from the system.....	209
Deleting DASD volumes from storage groups.....	209
Deleting storage groups.....	210
Deleting or moving system-managed data sets from the storage group.....	210
Preventing job failures.....	210
Deleting SMS classes.....	211
Converting volumes to SMS.....	212
Deleting DASD volume residual data.....	212
<b>Chapter 16. Recovering Storage Management Subsystem information.....</b>	<b>213</b>
Recovering control data sets.....	213
Recovering an SCDS.....	213
Recovering an ACDS.....	213
Recovering a COMMDS.....	214
Recovering from a systems failure in the SMS complex.....	215
Recovering the SMS address space.....	215
Canceling the SMS address space.....	215
<b>Chapter 17. Protecting the Storage Management Subsystem.....</b>	<b>217</b>
Identifying the resource owner and extracting the default classes.....	217
Protecting ISMF functions.....	217
Locating module names for ISMF applications.....	219
Locating module names for ISMF functions.....	219
Locating module names for line operators and commands.....	221
Protecting modules.....	226
Storage administration (STGADMIN) profiles in the FACILITY class or XFACILIT class.....	226
Command and keyword related profiles.....	227
Authority to activate a Storage Management Subsystem configuration.....	231
FIELD resource class.....	232
How authorization and protection of classes differ.....	232
Authorizing the storage and management classes.....	232
Protecting the control data sets.....	232
Authorizing for TSO/E and ISPF.....	233
Storage administrator authorization.....	233
<b>Chapter 18. Administering VSAM record-level sharing.....</b>	<b>235</b>
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.....	242
Defining the primary CF lock structure.....	244
Modifying the SYS1.PARMLIB IGDSMSxx member.....	248
Establishing authorization for VSAM RLS.....	249
Using dssTimeOut.....	249
Defining CF cache structures in the SMS base configuration.....	250
Defining storage classes for VSAM RLS.....	251
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.....	252
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.....	255
Changing IGDSMSxx parameters to support the coupling facility.....	256
Changing the state of coupling facility cache structures and volumes.....	259
Selecting data sets for coupling facility statistical monitoring.....	259
Recovering VSAM RLS processing.....	260
Recovering the CF lock structure.....	260
Recovering a CF cache structure.....	261
Recovering the SMSVSAM server address space.....	261
Recovering a sharing control data set.....	261
Falling back from VSAM RLS processing.....	262
Fallback rules and considerations.....	262
Fallback procedure.....	262
Terminating the SMSVSAM address space.....	264
Quiescing or enabling a secondary lock structure.....	264
Deleting a VSAM RLS lock structure.....	264
Displaying information about a secondary lock structure.....	264
<b>Chapter 19. Writing ACS routines.....</b>	<b>267</b>
Constants.....	267
Simple mask rules.....	268
Simple mask examples.....	268
Data set mask rules.....	268
Data set mask examples.....	268
Read-write variables.....	269
Using read-write variables with volume reference (VOL=REF).....	269
Using read-write variables with data set stacking.....	270
Read-only variables.....	270
Using read-only variables.....	281
Constraints when using read-only variables.....	283
Comparison operators.....	284
Comparison rules.....	285
Boolean expressions.....	285
Special functions.....	286
Determining Distributed FileManager/MVS data set creation requests.....	288
Statements.....	289
PROC statement.....	290
FILTLIST statement.....	290
SET statement.....	292
DO statement.....	292
IF statement.....	293
SELECT statement.....	294
EXIT statement.....	295
WRITE statement.....	295
END statement.....	297
LOGWRITES statement.....	297
Sample ACS routine.....	298
<b>Chapter 20. Quick reference to ISMF commands and line operators.....</b>	<b>299</b>
<b>Chapter 21. SETCACHE functions and device information.....</b>	<b>309</b>
Maintaining the media characteristics of a volume.....	309
Modifying DASD storage control characteristics.....	309
Modifying caching characteristics.....	310
Modifying duplexing characteristics.....	311
Modifying destaging characteristics.....	311
Submitting jobs.....	311
Providing remote authorization codes.....	312

<b>Chapter 22. Space utilization and capacity planning.....</b>	<b>313</b>
Using the data collection application.....	313
<b>Chapter 23. Using data set separation.....</b>	<b>317</b>
Overview.....	317
Syntax for creating a data set separation profile.....	317
Deprecated syntax.....	318
Writing comments.....	319
Indicating continuation and termination.....	319
Example.....	320
Volume selection for data set separation by volume.....	320
Data set requirements for a data set separation profile.....	320
Creating multiple data set separation profiles.....	320
Using data set separation with generation data groups and striping.....	321
Environmental conditions affecting data set separation.....	321
Factors affecting code path length.....	322
<b>Chapter 24. Using NaviQuest.....</b>	<b>323</b>
Starting NaviQuest.....	323
Terminology.....	323
Protecting NaviQuest.....	325
Methods for collecting test data.....	325
ISMF lists.....	325
DCOLLECT data.....	325
SMF data.....	325
VMA data.....	326
Data classification methodology.....	326
Testing ACS routines.....	327
Setting up the test environment.....	327
Testing procedures.....	328
NaviQuest testing scenario.....	334
Activating an ACS routine.....	335
Performing storage administration tasks in batch.....	335
How to run storage administration tasks in batch.....	335
Using the sample JCL for batch.....	336
NaviQuest CLISTS and REXX EXECs.....	341
Storage group volume add/Delete: ACBQBAI9.....	361
ACS routine translate: ACBQBAO1.....	363
SCDS validation: ACBQBAO2.....	366
Test ACS routines: ACBQBAIA.....	367
Generate test cases from ISMF-saved data set lists: ACBQBAG3.....	368
Generate test cases from DCOLLECT data: ACBQBAG1.....	369
Generate test cases from SMF data.....	369
Generate test cases from VMA extract data: ACBQBAO3.....	370
ACS test listings comparison: ACBQBAC1.....	371
Enhanced ACS test listing: ACBQBAX1.....	372
Update test cases with expected results: ACBQBAU1.....	373
Generate report from ISMF-saved data set list: ACBQBAR1.....	373
Generate report from ISMF-saved DASD volume list: ACBQVAR1.....	376
Generate report from ISMF-saved tape list: ACBQBAR4.....	379
Generate data set report from DCOLLECT data: ACBQBAR7.....	381
Generate model commands from ISMF-saved list: ACBQBAM1.....	383
Creating SMS online reports.....	384
Creating data set reports from saved ISMF lists.....	385
Creating volume reports from saved ISMF lists.....	385
Creating customized tape reports from saved ISMF lists.....	386

Data set report from DCOLLECT data.....	386
DASD volume report from DCOLLECT data.....	386
SMS configuration report from DCOLLECT data.....	387
Additional storage administration functions.....	387
QSAVE and QRETRIEV commands: saving and retrieving ISMF selection criteria.....	387
Creating model commands.....	388
COPYFILT macro: COPYLIB facility for FILTLISTs.....	390
<b>Appendix A. Sample batch job for CICS definitions.....</b>	<b>393</b>
<b>Appendix B. Accessibility.....</b>	<b>395</b>
<b>Notices.....</b>	<b>397</b>
Terms and conditions for product documentation.....	398
IBM Online Privacy Statement.....	399
Policy for unsupported hardware.....	399
Minimum supported hardware.....	399
Programming interface information.....	400
Trademarks.....	400
<b>Index.....</b>	<b>401</b>



---

# Figures

1. ISMF Primary Option Menu for Storage Administrators.....	2
2. System Grouping Concept.....	8
3. Relationship among SCDSs and ACDSs in an Installation.....	12
4. Formula for Deriving Optimum Number of Stripes.....	30
5. Defining Volume System Status.....	43
6. Defining Dummy Storage Group Volume Serial Numbers.....	45
7. Allocation of Primary and Secondary Space for Multivolume Data Sets.....	96
8. Allocation of Primary and Secondary Space for Multivolume Data Sets with Different Primary Allocations on Each Volume.....	97
9. Editing an Aggregate Group Selection Data Set- Example.....	140
10. Pool Storage Groups and Copy Pool Backup Storage Groups.....	143
11. Browsing the Results of an Unsuccessful ACS Translation.....	153
12. Browsing the Results of an ACS Routine Validation.....	155
13. Example of REF=ST Values when Using VOL=REF.....	164
14. Example of REF=NS Values when Using VOL=REF.....	164
15. Example of Failing VOL=REF Values That Are Not Valid.....	164
16. Example of a Storage Class ACS Routine for Read-Only Variables.....	166
17. Example of a Storage Group ACS Routine for Read-Only Variables.....	166
18. Example of a Storage Class ACS Routine Specifying Both Volume Reference and Unit Affinity.....	168
19. Displaying Information about the Active Configuration.....	186
20. Displaying Trace Information.....	186
21. DISPLAY SMS,STORGRP Command Syntax.....	187
22. Displaying Storage Group Status Information.....	187

23. Displaying Storage Group Volume Status Information.....	188
24. Displaying Cache and DASD Fast Write Information.....	189
25. Displaying Status Information for Individual DASD Volumes.....	192
26. DEVSERV S,430 Output.....	193
27. DEVSERV QPAVS Display Content.....	195
28. DS QP,D123,3 Output.....	196
29. DS QP,D2FF,VOLUME Output.....	196
30. DS QP,D222,VOLUME Output.....	196
31. DS QP,D6FF,UNBOX Output.....	196
32. DS QP,D5EF,UCB Output.....	197
33. DS QP,SSID=1401 Output.....	197
34. DS QP,D345 Output.....	197
35. DS QP,D621 Output.....	198
36. DS QP,D6F4 Output.....	198
37. DS QP,F60 Output.....	198
38. DS QP,E27F,HPAV Output.....	198
39. DS QP,E200,HPAV Output.....	199
40. DS QP,E27F,VOLUME Output.....	199
41. DEVSERV P,F4A Output.....	199
42. DS QD,F41,RDC,DCE Output.....	200
43. Example of Global Connectivity in a Parallel Sysplex.....	237
44. Example of Insufficient Connectivity in a Parallel Sysplex.....	238
45. Example for creating sharing control data sets using IDCAMS.....	240
46. Example of CF Cache Structure Definition in Base Configuration.....	251
47. DISPLAY SMS command.....	253

48. Example of Constraints when Using the &ALLVOL and &ANYVOL Read-Only Variables.....	287
49. Example Showing &JOB Read-Only Variable.....	288
50. Example Showing &PGM Read-Only Variable.....	289
51. Example Showing &USER Read-Only Variable.....	289
52. Example of a FILTLIST Statement.....	291
53. Using the INCLUDE and EXCLUDE.....	291
54. Example of a DO Statement.....	293
55. Examples of IF Statements.....	293
56. Example of a SELECT Statement.....	295
57. Example of an EXIT Statement.....	295
58. Example of an END Statement.....	297
59. Production ACS Routine for Storage Class.....	298
60. Typical Data Types.....	326
61. TSO Data Type with Data Subtypes.....	326
62. Sample JCL for ACBJBAOS.....	367
63. Sample JCL for ACBJBAIC.....	368
64. Sample JCL for ACBJBAG2.....	369
65. Sample JCL for ACBJBAG1.....	369
66. Sample JCL for ACBJBAI1.....	370
67. Sample JCL for ACBJBAOW.....	370
68. Sample JCL for ACBJBAC1.....	371
69. Sample JCL for ACBJBAU2.....	373
70. Sample JCL for ACBJBAM1.....	384
71. Copy FILTLISTs Entry Panel.....	392





---

# Tables

1. Summary of media selection for object storage.....	32
2. Volumes Listed in a Specified Storage Group.....	53
3. Comparing Retention Period Attributes.....	72
4. MSR Capabilities.....	87
5. D/T3990 SMS Cache Candidate Tokens for Sequential and Direct Requests.....	89
6. Combinations for Requesting Point-in-Time Copy Devices.....	94
7. Volume Selection Preference Attributes.....	101
8. Applying Data Class Attributes to Record Organization (Recorg).....	129
9. Results of Output Listing Disposition.....	152
10. Allocation, IMPORT, and COPY Conditions.....	158
11. RESTORE, RECALL, RECOVER, CONVERTV, and FORCENONSMS Conditions.....	160
12. Values for &UNIT ACS Read-Only Variable.....	165
13. AFFing DD Volser Values.....	167
14. DFSMSrmm Read-Only Variables.....	170
15. Module Names for ISMF Applications.....	219
16. Module Names for ISMF Functions.....	219
17. Module/CLIST Names for ISMF Line Operators, Part 1.....	221
18. Module/CLIST Names for ISMF Line Operators, Part 2.....	222
19. Module/CLIST Names for ISMF Line Operators, Part 3.....	223
20. Module/CLIST Names for ISMF Commands.....	225
21. Effect of MAXSYSTEM Value on Lock Table Entry Size.....	244
22. CF Lock Structure Sizing Examples.....	244
23. Using Read-Write Variables in ACS Routines.....	269

24. Read-Only Variables.....	271
25. Indexing Functions for Read-Only Variables.....	286
26. Read-Write Variable Assignments.....	292
27. ISMF Commands.....	300
28. ISMF Line Operators.....	303
29. PCU-Level Syntax Examples.....	319
30. SYS1.SACBCNTL Sample JCL Library Member List.....	337
31. CLISTs and REXX EXECs for Storage Administration Tasks.....	341
32. Using ACBQBAl2, ACBJBAI7, and ACBJBAOU Parameters.....	345
33. ACBQBAl4 Parameters.....	349
34. ACBQBAlD Parameters.....	353

## About this document

---

This document introduces system programmers and storage administrators to the IBM Storage Management Subsystem (SMS) and storage management concepts. It describes how to define, initialize, and maintain SMS and how to manage storage with the IBM Interactive Storage Management Facility (ISMF).

This document is intended for system programmers and storage administrators like you who manage storage under DFSMS. 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.

This document is specific to DFSMSdfp. For information about DFSMSshm and DFSMSdss, see [z/OS DFSMSshm Storage Administration](#) and [z/OS DFSMSdss Storage Administration](#).

For information about the accessibility feature of z/OS, for users who have a physical disability, see [Appendix B, “Accessibility,”](#) on [page 395](#).

## Required product knowledge

---

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

- Storage management concepts
- ISMF applications
- DFSMS Data Set Services (DFSMSdss)
- DFSMS Hierarchical Storage Manager (DFSMSshm)
- Device Support Facility (ICKDSF)
- Object access method (OAM)

## 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 document 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 | 0}`

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.*

## z/OS information

---

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

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



## How to provide feedback to IBM

---

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. For more information, see [How to send feedback to IBM](#).





## Summary of changes

---

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

**Note:** IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under [IBM z/OS Product Documentation Update Policy](http://www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy) ([www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy](http://www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy)).

## Summary of changes for z/OS 3.2

---

The following content is new, changed, or no longer included in z/OS 3.2.

### New

The following content is new.

#### September 2025 release

- None.

### Changed

The following content is changed.

#### September 2025 release

- Updated NaviQuest. For more information, see [“Generate report from ISMF-saved DASD volume list: ACBQVAR1” on page 376](#), [“Sample JCL \(ACBJBAOF\) for generating a report from ISMF-saved DASD volume list” on page 377](#), [“Sample JCL \(ACBJBAI4\) for generating a volume list ” on page 352](#) and [“Generate a DASD volume list: ACBQBAI4” on page 349](#)

### Deleted

The following content is deleted.

#### September 2025 release

- None.

## Summary of changes for z/OS 3.1

---

The following content is new, changed, or no longer included in z/OS 3.1.

### New

The following content is new.

#### June 2024 refresh

Profile STGADMIN.SMS.DATASET.ENCRYPT.PREFERRED is added. For more information, see [“Command and keyword related profiles” on page 227](#). (APAR OA66122, which also applies to z/OS 2.5)

### Changed

The following content is changed.

### **December 2024 refresh**

- “[Values for defining a pool storage group](#)” on page 35 attribute “[Total Space Alert Threshold %](#)” on page 39 is updated. (OA66369: NEW FUNCTION - SMS Support Enhancement for DS8K Compression ([www.ibm.com/support/pages/apar/OA66369](http://www.ibm.com/support/pages/apar/OA66369)), which also applies to z/OS V2R5 and newer.)
- “[Command and keyword related profiles](#)” on page 227 is updated. (OA66738, which also applies to z/OS V2R5 and newer.)

### **October 2024 refresh**

- “[Defining use of zHyperLinks](#)” on page 99 is updated. (APAR OA66438 ([www.ibm.com/support/pages/apar/OA66438](http://www.ibm.com/support/pages/apar/OA66438)), which also applies to z/OS V2R5)

### **August 2024 refresh**

- “[Selecting volumes for control data sets](#)” on page 15 is updated with a restrictions note.

### **May 2024 refresh**

- Selecting volumes for control data sets is updated. For more information, see “[Selecting volumes for control data sets](#)” on page 15.

### **September 2023 release**

- “[Values for defining a pool storage group](#)” on page 35 is updated. (OA64835)
- Note added that the Size Less Than or Equal To and Size Greater Than attributes are only used during space management. For more information, see “[Defining management class migration attributes](#)” on page 73.
- “[WRITE statement](#)” on page 295 is updated.

---

# Chapter 1. Introducing the Storage Management Subsystem

This topic introduces the Storage Management Subsystem (SMS). 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. It also 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
  - 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 [Chapter 24, “Using NaviQuest,”](#) on page 323 for more information on the DFSMS NaviQuest tool.

Figure 1 on page 2 shows the ISMF Primary Option Menu for storage administrators.

```

Panel  Help
-----
                ISMF PRIMARY OPTION MENU - z/OS DFSMS V2 R3
Selection or Command ==>

0  ISMF Profile                - Specify ISMF User Profile
1  Data Set                    - Perform Functions Against Data Sets
2  Volume                      - Perform Functions Against Volumes
3  Management Class            - Specify Data Set Backup and Migration Criteria
4  Data Class                  - Specify Data Set Allocation Parameters
5  Storage Class               - Specify Data Set Performance and Availability
6  Storage Group              - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection   - Specify ACS Routines and Test Criteria
8  Control Data Set           - Specify System Names and Default Criteria
9  Aggregate Group            - Specify Data Set Recovery Parameters
10 Library Management          - Specify Library and Drive Configurations
11 Enhanced ACS Management    - Perform Enhanced Test/Configuration Management
C  Data Collection             - Process Data Collection Function
G  Report Generation          - Create Storage Management Reports
L  List                        - Perform Functions Against Saved ISMF Lists
P  Copy Pool                  - Specify Pool Storage Groups for Copies
R  Removable Media Manager     - Perform Functions Against Removable Media
S  Network Connection         - Specify Network Connection Attributes
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
- Enhanced ACS Management
- Copy Pool
- Network Connection

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 [Chapter 22, “Space utilization and capacity planning,”](#) on page 313.

## Customizing ISMF

There might be times when you want to customize the interactive storage management facility (ISMF). This topic lists the libraries that you can customize and some restrictions. For more information, see [z/OS DFSMS Using the Interactive Storage Management Facility](#), which describes ISMF restrictions and customization with DFSMSdss.

## ISMF libraries that can be customized

You can customize the following ISMF libraries:

### Panel Library

ISMF allows you to make the following changes:

- Change the initial priming values that ISMF ships
- Change the default values for data entry panels
- Provide additional restrictions to values that are entered for certain fields on panels
- Remove fields from functional panels
- Change highlighting and color
- Change the format of the panel
- Modify existing functional panel text and help text
- Add new fields to panels
- Add new panels.

### Message Library

You can modify existing messages and add new messages.

### Skeleton Library

You can modify the job skeletons for ISMF commands and line operators.

### Table Library

You can modify the Interactive System Productivity Facility (ISPF) command tables.

### Load Library

You can modify the ISMF command and line operator tables. The tables are contained in nonexecutable CSECTs in the load library.

### CLIST Library

You can modify the options on the CLIST CONTROL statement.

## Restrictions to customizing ISMF

General restrictions for ISMF customization include the following:

1. Before changing anything, you should make a backup copy of ISMF. Keep this unmodified version of the product for diagnostic purposes. IBM support and maintenance is provided only for the unmodified version of ISMF.
2. Do not delete or rename any of the parts of ISMF. Deleting or renaming severely affects processing and can cause ISMF to fail.
3. ISMF is copyrighted. Under the IBM licensing agreement you may modify ISMF for your own use. You may not, however, modify it for commercial resale.

## 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 types of classes and groups:

**Storage group**

Use this 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**

Use this to define different levels of migration, backup, class transition 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 a data set or an object characteristic that might trigger a class transition.

**Storage class**

Use this 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**

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

**Aggregate group**

Use this to define groups of data sets for the purpose of backing up or recovering all data sets in a group in a single operation.

**Copy pool**

Use this to define a pool of storage groups to be processed collectively for fast replication operations.

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).

## **A valid minimal SMS configuration**

To define a valid minimal SMS configuration, the configuration must contain:

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

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

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

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

See *z/OS DFSMS Implementing System-Managed Storage* for more information on defining a minimal SMS configuration.

## The SMS construct naming convention

The general SMS construct naming convention must:

- Use upper-case alphabetic, numeric, and/or national (\$, @, #) characters only. The alphabetic characters are the 26 English letters.
- Consist of a name that must begin with an alphabetic or national character
- Not contain any leading or embedded blanks.
- Contain eight characters or less

Exceptional naming for the following SMS constructs:

- Copy pool:
  - Name can also contain an underscore symbol.
  - The maximum length is 23.
- Copypool backup storage group:
  - Name must begin with an alphabetic character only.

## Preparing for and implementing the Storage Management Subsystem

---

To prepare for and implement a storage management policy, you perform these steps:

1. Prepare for SMS, as described in [Chapter 2, “Preparing for the Storage Management Subsystem,”](#) on page 11. This includes:
  - a. Allocating control data sets, which contains information used by SMS.
  - b. Modifying and creating SYS1.PARMLIB members to identify SMS to all the systems in the SMS complex. These new members take effect when you IPL.
  - c. Establishing 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.
2. Define the base configuration, which identifies the systems within the SMS complex. See [Chapter 3, “Creating the base configuration,”](#) on page 19.
3. 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. See [Chapter 4, “Defining storage groups,”](#) on page 25, [Chapter 8, “Defining management classes,”](#) on page 67, [Chapter 9, “Defining storage classes,”](#) on page 83, [Chapter 9, “Defining storage classes,”](#) on page 83, [Chapter 11, “Defining aggregate groups,”](#) on page 137 and [Chapter 11, “Defining aggregate groups,”](#) on page 137. Define optical libraries and drives and tape libraries, if you have them. 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*.
4. Define ACS routines to assign the SMS classes and storage groups. Then test the routines. See [Chapter 13, “Defining ACS routines,”](#) on page 149.
5. 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.
6. Activate the valid SMS configuration. See [Chapter 14, “Activating Storage Management Subsystem configurations,”](#) on page 179.

## 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:

1. 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, z/OS Security Server RACF® security and global shared resources (GRS) serialization, work only 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.
2. 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.
3. In a JES3 environment, system group name is not supported due to the fact that the system features and resources that are used to determine on which eligible system a job will be run are only applicable to each system, not to a system group.

## 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.

## 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:

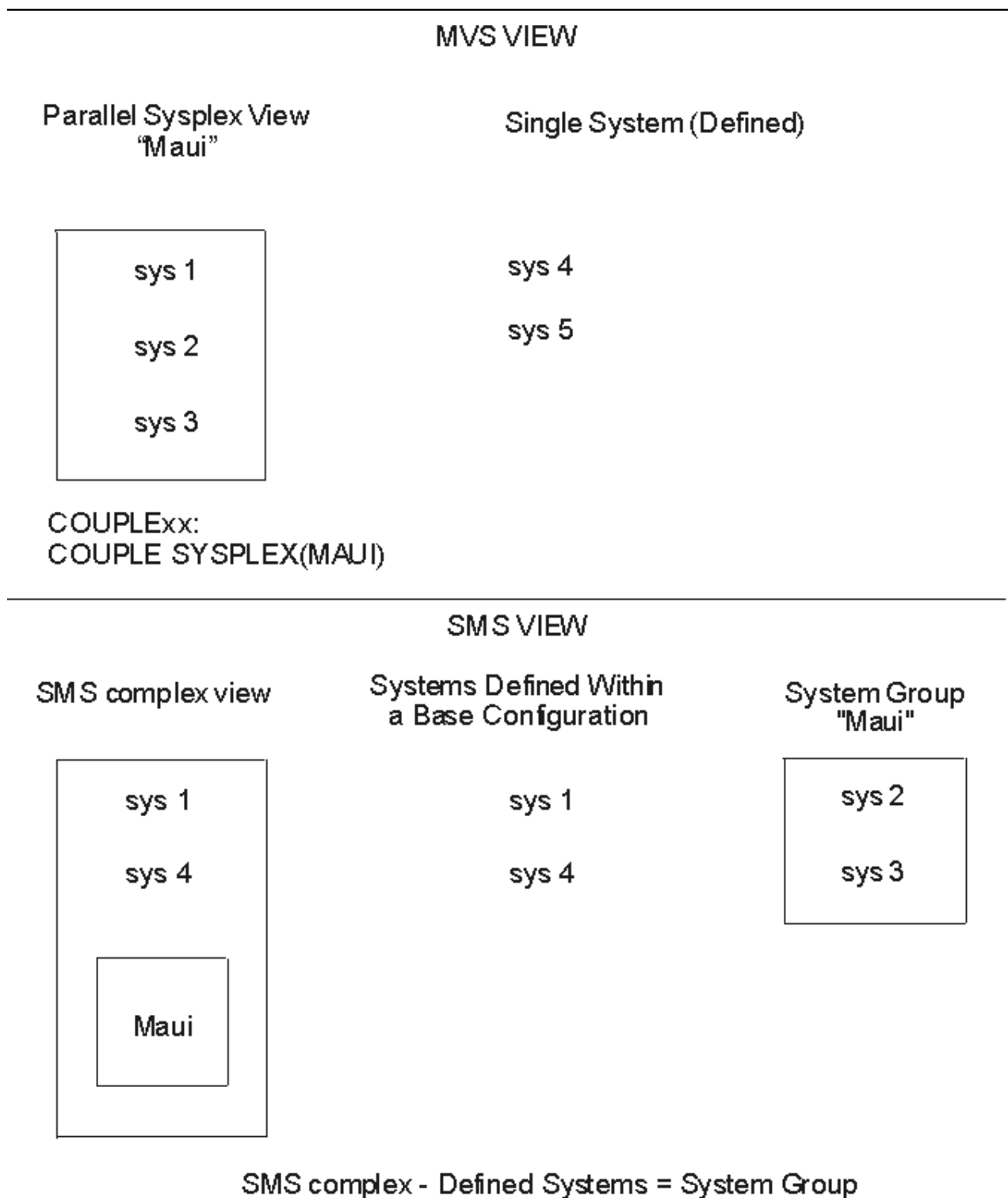
- Running the same SMS configuration



- 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 2 on page 8](#) provides a visual representation of the system grouping concept.



DA6S2015

*Figure 2. System Grouping Concept*

In Figure 2 on page 8, 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 Upgrade Workflow\*](#) 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, 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. See [“Allocating control data sets” on page 11](#).
- Define SMS as trusted to RACF. This generally allows the SMS address space to bypass RACF authorization checking and so access the SMS configuration data sets.
- Modify SYS1.PARMLIB, which contains three members that direct the initialization and activation of SMS. See [“Modifying the SYS1.PARMLIB data set” on page 17](#).
- Establish access to the ISMF Primary Option Menu for Storage Administrators (shown in [Figure 1 on page 2](#)).

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

For more information about planning and implementing SMS, see [z/OS DFSMS Implementing System-Managed Storage](#).

For information about planning and preparing for SMS with object support, optical libraries, or tape libraries, 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](#).

---

### Allocating control data sets

Before you can activate an SMS configuration, allocate the SMS control data sets 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, copy pool, 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).

#### Restrictions:

1. Do not name any of your SMS control data sets the single word ACTIVE. SMS uses the single word ACTIVE as a reserved word that indicates active configuration residing in the SMS address space. Naming an SMS control data set 'ACTIVE' results in errors.
2. SMS control data sets cannot be accessed from a volume that is configured as read-only.
3. Do not allocate control data sets as extended format VSAM linear data sets.
4. Do not allocate control data sets as encrypted VSAM linear data sets.
5. Do not catalog the SMS control data sets (SCDS, ACDS, COMMDS) in a user catalog that is RLS-managed.

### 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 number of SCDSs, but only one can be put in the ACDS. [z/OS MVS Initialization and Tuning Guide](#) 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.

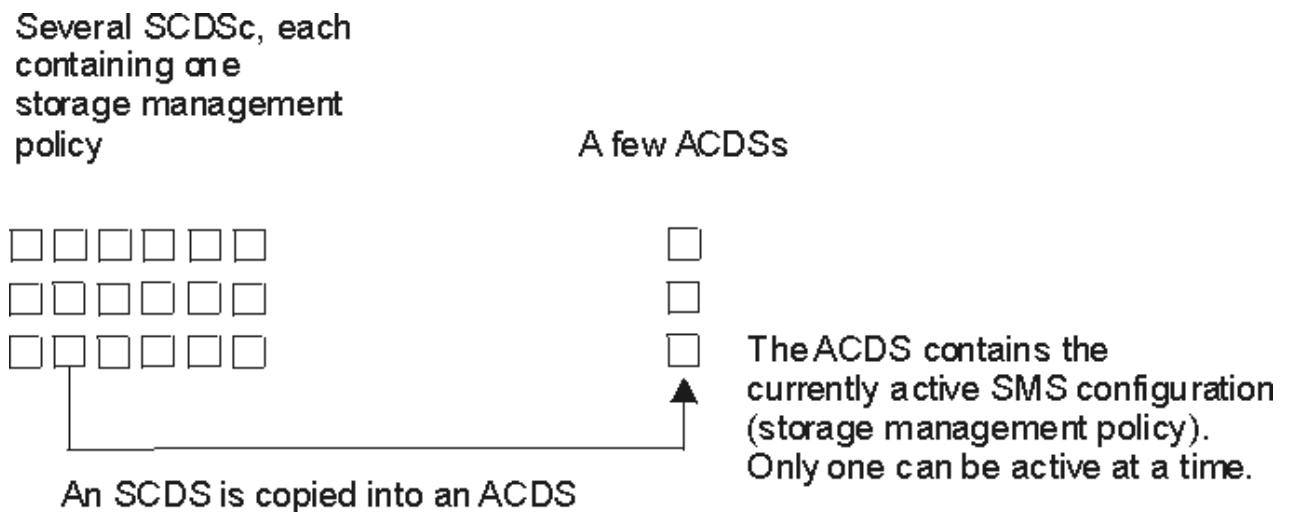
**Tip:** You can save the current ACDS as an SCDS using the SETSMS SAVESCDs command. You can also create an ACDS from a SCDS with the SETSMS COPYSCDS command. For more information, see “Parameters of the SETSMS operator command” on page 182.

**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 3 on page 12 shows the relationship among SCDSs and ACDSs in an installation.



DA 6S2014

Figure 3. 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 COMMDs, 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 and base configuration information, SMS class, aggregate group, storage optical library and drive, and other SMS constructs.

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

```
size (bytes) = 3 * (170000 +  
    (11136 * CP) +  
    ( 2728 * SG) +  
    (  312 * SC) +  
    (  736 * MC) +  
    (  552 * DC) +  
    ( 2224 * VOL) +  
    (  832 * AG) +  
    ( 2240 * LIB) +  
    ( 2208 * DRV) +  
    (  520 * DST) +  
    (   56 * AB) +  
    (  184 * CS) +  
    (   64 * CSS) +  
    (   40 * LK) +  
    (   32 * LS) +  
    ( 1280 * CL) )
```

where:

### **170000 bytes**

represent the estimated size of the base configuration, ACS routines (estimated average 40 KB for each), headers, and other fixed data fields.

### **CP**

The estimated number of Copy Pools.

### **SG**

The estimated number of storage groups.

### **SC**

The estimated number of storage classes.

### **MC**

The estimated number of management classes.

### **DC**

The estimated number of data classes.

### **VOL**

The estimated number of DASD volumes in the SMS complex to be managed by SMS.

### **AG**

The estimated number of aggregate groups.

### **LIB**

The estimated number of optical and tape libraries in the SMS complex to be managed by SMS.

### **DRV**

The estimated number of optical drives in the SMS complex to be managed by SMS.

### **DST**

The estimated number of lists of destination names in the management class.

### **AB**

The estimated number of aggregate backup groups in the management class.

### **CS**

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

### **CSS**

The estimated number of cache structures in the base configuration (one cache set can have up to eight cache structures).

**LK**

The estimated number of lock sets in the base configuration (for record-level sharing (RLS) only).

**LS**

The estimated number of lock structures in the base configuration (one lock set can have only 1 lock structure).

**CL**

The estimated number of cloud classes.

Due to the format in which control data sets are written and for recovery purposes, the estimated size of a control data set must have room for three configurations. For example, suppose you have the following configuration:

- Five storage groups
- Four management classes
- Two storage classes
- Five data classes
- Two aggregate groups
- Six optical libraries
- 24 optical drives
- 40 DASD volumes

With this configuration, the equation yields a value of 1,044,912 bytes (~ 1 MB) for control data set allocation.

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 can be too small because the storage requirements for classes and groups can be different between releases or new classes, groups, or other items can be added. It is recommended that secondary space be specified when allocating the SCDS and ACDS to ensure extends can occur when new classes, groups, and others are added or sizes of these classes, groups, and others increase in size.

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 can 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):

$\text{COMMDS size (bytes)} = 8192 + (588 * \text{VOL})$

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 volumes serials provided, or derived from the data class assigned to the data set (the greater value of volume count or dynamic volume count). For more information, see [Chapter 10, “Defining data classes,” on page 117](#) for more information. For SMS control data sets, the volume count or dynamic volume count must be less than or equal to the number of online volumes the data set resides on, otherwise, you might receive messages IEF244I and IEF877E.

### Restriction:

- You can have your SMS-managed control data sets (SCDS, ACDS, and COMMDs) defined as multivolume. When you define these data sets as multivolume, you must assign a storage class with the guaranteed space attribute. This is because the control data sets are opened in the SMS address space as a non-SMS managed. If guaranteed space is not specified in the storage class, your SMS-managed CDS cannot be extended to a new volume if no candidate volumes with space are already allocated.
- You cannot use dynamic volume count in the data class to allow extends to new volumes for any SMS-managed CDS. This is true when the SMS-managed CDS (SCDS, ACDS, COMMDs) is opened under the SMS address space such as activation or by using the SETSMS commands (COPYSCDS, SAVEACDS).

If you have a multivolume SCDS that 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 by using the SETSMS command. For more information about the SETSMS command, see [“Changing Storage Management Subsystem parameters” on page 182](#)

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 that can be 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) REUSE)
/*
```

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 Commands](#) 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) REUSE)
/*
```

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 16, “Recovering Storage Management Subsystem information,” on page 213 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 19. 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) REUSE)
/*
```

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 16, “Recovering Storage Management Subsystem information,” on page 213 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.

### Recommendations:

1. Specify the REUSE option when defining the COMMDS data set to help avoid space problems (SMS reason code 6068) during subsequent COMMDS updates or IMPORT/EXPORT functions.
2. The COMMDS and ACDS must be accessed from all systems in the complex simultaneously.

## Recommendations

1. Specify the REUSE option when defining the COMMDS data set to help avoid space problems (SMS reason code 6068) during subsequent COMMDS updates or IMPORT/EXPORT functions.
2. Use SHAREOPTIONS(3,3) when allocating an ACDS or COMMDS. This allows full authority to read from and write to the ACDS or COMMDS from any system. The ACDS and COMMDS must be accessed from all systems in the complex simultaneously. If SMS detects a lower value for SHAREOPTIONS when the CDS is activated, SMS attempts to alter the value to SHAREOPTIONS(3,3).

## Defining SMS as trusted to RACF

---

Defining SMS as trusted to RACF generally allows the SMS address space to bypass RACF authorization checking and so access the SMS configuration data sets.

Regardless of what release level your system is running, assigning the TRUSTED attribute to the SMS address space is required, because SMS accesses a variety of unpredictably named data sets, in particular the SMS configuration data sets. SMS also supports some automatic functions to maintain the SMS configuration data sets with the appropriate attributes to aid in accessing the SMS configuration data sets more efficiently. For example, when SMS determines that the REUSE option is not specified when the CDS is activated, SMS automatically attempts to alter the CDS to have the REUSE option.

For details about defining SMS as trusted to RACF, refer to the topic about [Associating started procedures and jobs with user IDs in \*z/OS Security Server RACF System Programmer's Guide\*](#), and the topic about [Using started procedures in \*z/OS Security Server RACF Security Administrator's Guide\*](#).

## Modifying the SYS1.PARMLIB data set

---

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

For information about how to create an IGDSMSxx member in SYS1.PARMLIB and how to define SMS to z/OS through IEFSSNxx, see [z/OS MVS Initialization and Tuning Reference](#).

## Initializing SMS through the IGDSMSxx member

Every SMS system must have an IGDSMSxx 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 IGDSMSxx 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.

## 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 IGDSMSxx 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 179](#). 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 information.

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 17, “Protecting the Storage Management Subsystem,” on [page 217](#) 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 18, “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 242](#)
- [“Defining the primary CF lock structure” on page 244](#)
- [“Modifying the SYS1.PARMLIB IGDSMSxx member” on page 248](#)
- [“Establishing authorization for VSAM RLS” on page 249](#)

---

## 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 topic 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 8, “Defining management classes,”](#) on page 67.

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

**Before you begin:** You must allocate an SCDS.

Perform the following steps to define a base configuration.

1. Select option 8, Control Data Set, from the ISMF Primary Option Menu for Storage Administrator. This displays the CDS Application Selection panel.

- 
2. Supply values on the CDS Application Selection panel:

**CDS name**

This is the name of the SCDS that is to contain the base configuration. ISMF primes the CDS Name field with the last used SCDS name

**Option**

Select option 2, Define. This displays the SCDS Base Define panel.

You can also:

- See the base configuration, by selecting option 1, Display. 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.
- Alter an SCDS base configuration that you defined previously, by selecting option 3, Alter. The Alter panel contains the same fields as the Define panel.

- 
3. Supply values on the SCDS Base Define panel.

Use the DOWN command to view the second page of the panel.

**Description**

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

**Default Management Class**

See [“Specifying the default management class” on page 21.](#)

**Default Unit**

See [“Specifying the default unit” on page 21.](#)

**Default Device Geometry**

See [“Specifying the default device geometry” on page 21.](#)

**DS Separation Profile (Data Set Name)**

See [“Specifying the DS separation profile” on page 21.](#)

**Systems and System Groups in the SMS Complex**

See [“Specifying systems and system groups in the SMS complex” on page 22.](#)

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

When you have supplied all of the values, use the END command to save them and return to the CDS Application Selection panel.

- 
4. Validate and activate the SCDS, as desired. See [“Validating the SCDS” on page 20](#) and [“Activating the SCDS” on page 20.](#)
- 

**Requirements and Restrictions:**

1. The SMS configuration 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 11](#)). 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.

2. 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.

**Validating the SCDS**

To validate the SCDS or any of the ACS routines, select option 4, Validate, from the CDS Application Selection panel. This displays the status of the SCDS.

You should validate an SCDS before you activate it. 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.

[“Validating an entire SCDS” on page 155](#) lists conditions that cause an SCDS to be not valid. [“Validating ACS routines or an entire SCDS” on page 153](#) provides more information about the validate option.

**Activating the SCDS**

To activate the SCDS, select option 5, Activate, from the CDS Application Selection panel.

You must activate the configuration for any changes made to the SCDS to take effect.

## 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) or volume level, from all the other data sets in the same group. This reduces the effects of single points of failure. Using the volume level reduces I/O contention.

To use 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 Chapter 23, [“Using data set separation,”](#) on page 317 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 quotation marks. For data set names without quotation marks, ISMF will add the TSO user ID prefix of the person who is defining or updating the base configuration.

**Recommendation:** Use a data set name that contains quotation marks.

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.

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

Figure 2 on page 8 provides a visual representation of the system grouping concept.

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 6.

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.

The system name must match the value of the SYSNAME parameter in the IEASYSy member of the SYS1.PARMLIB used on that system. The system group name must match the value of the SYSPLEX parameter in the COUPLExx (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.

**Restriction:** In a JES3 environment, system group name is not supported due to the fact that the system features and resources that are used to determine on which eligible system a job will be run are only applicable to each system, not to a system group.

- 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 215](#) 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 sets 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 242](#) for more information on defining CF cache structures. See [“Defining the base configuration” on page 19](#) for more information on defining the base configuration.

### Defining cache sets

**Before you begin:** You must define Cache structures to z/OS in a CFRM policy.

Perform the following steps to define a cache set and specify the cache structures associated with it.

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

- 
2. Supply values on the Control Data Set (CDS) Application Selection panel:

**CDS Name**

This is the name of the SCDS that is to contain the base configuration for VSAM RLS. The CDS Name field is primed with the last used SCDS.

**Option**

Select option 7, Cache Update. This displays the CF Cache Set Update panel.

- 
3. Supply values on the CF Cache Set Update panel:

**Cache Set**

This is the name of a cache set. You can define up to 256 cache set names.

**CF Cache Structure Names**

These are 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.

## Defining lock sets

Perform the following steps to define a lock set and specify the lock set structure associated with it.

1. Select option 8, Control Data Set, from the ISMF Primary Option Menu for Storage Administrators. This displays the CDS Application Selection panel.

- 
2. Supply values on the CDS Application Selection panel:

**CDS Name**

This is the name of the SCDS that is to contain the base configuration for VSAM RLS. ISMF primes the CDS Name field with the last used SCDS name.

**Option**

Select option 9, Lock Update. This displays the CF Lock Set Update panel.

- 
3. Supply values on the CF Lock Set Update panel:

**Lock Set**

This is the name of a lock set. You can define up to 256 lock set names.

**CF Lock Structure**

This is the name of the CF lock structure associated with the lock set. You can specify a CF lock structure for each lock set.

## 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 topic 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, copy pool backup, 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, the actual, physical paths connecting systems to DASD volumes must 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. The physical connection must exist.

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 27.

### 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\*](#) 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. The object storage hierarchy can consist of:

- Disk sublevel 1, which is associated with Db2 tables on a direct access storage device (DASD). The following object storage Db2 tables provide disk sublevel 1 storage for objects:
  - 4 KB storage table
  - 32 KB storage table
  - LOB storage structure
  - Cloud storage
- Disk sublevel 2, which is associated with an NFS or zFS file system.
- Tape sublevel 1 volumes associated with a tape library device (SMS-managed, library-resident tape volumes), and tape volumes outside of a library device (non-SMS-managed, shelf-resident tape volumes)
- Tape sublevel 2 volumes associated with a tape library device (SMS-managed, library-resident tape volumes), and tape volumes outside of a library device (non-SMS-managed, shelf-resident tape volumes)
- Optical volumes inside a library device (SMS-managed, library-resident optical volumes), and optical volumes outside of a library device (SMS-managed, shelf-resident optical 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.

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.

You can specify two timing attributes for object and object backup storage groups: cycle start time and cycle end time. These attributes define a window of time 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.

You can also specify an OSMC processing system name. This attribute identifies the OAM system in the sysplex that performs OSMC processing for the object or object backup storage group. Specifying the OSMC processing system name prevents multiple systems from trying to process the same storage group at the same time. This is particularly useful when you use the cycle start time attribute to automatically start storage group processing.

You can also specify the Db2 subsystem identification. Each Object OAM instance in a multiple OAM configuration is associated with a unique Db2 subsystem. A given object or object backup storage group can be associated with one or more Object OAM instances. The Db2 identifier specified in the storage group definition determines which Object OAM instances in a multiple OAM configuration are associated with the object or object backup storage group. Providing a specific Db2 identifier associates only one Object OAM instance (the one associated with the specified Db2 identifier) with the object or object backup storage group. Wildcard characters (%) can be included in the Db2 identifier to associate the storage group with more than one OAM instance.

To protect objects in an object storage group from deletion:

- Specify deletion-protection mode. Objects cannot be deleted prior to their expiration date.
- Specify retention-protection mode. Objects cannot be deleted prior to their expiration date, and the expiration date can never be changed to an earlier date.

For more information, see [“Defining an object storage group” on page 46](#).

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 that are listed in the object backup storage group definition.

If no tapeunitname is specified and backuptier is not specified or specified with TAPEOPT in the CBROAMxx member of PARMLIB for this object backup storage group, OAM stores the backup copies to optical. Otherwise, OAM stores the backup copy or copies of the object using the backuptier specified in the CBROAMxx PARMLIB member.

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 (Db2 table, file system, optical, tape, or cloud). Objects can be written to tape using the media type and tape unit name defined in the SETOAM statements in the CBROAMxx 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 CBROAMxx 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 CBROAMxx 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.

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 an OAM Db2 object directory. A collection is described by a collection name entry in the OAM Db2 Collection Name Table.

OAM provides two primary interfaces for processing objects: the application programming interface (OSREQ) and the OAM Storage Management Component (OSMC). Both address objects that use the object's collection name and object name that include those cases where the object must be identified to the ACS routines for the STORE (OSREQ), CHANGE (OSREQ), and CTRANS (OSMC class transition) environments. See [\*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.

**Note:** The information provided by LISTVOL differs from that provided by LISTSYS, because the information for the two commands comes from different sources.

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 40](#) 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.

If you use the DFSMSrmm storage group support for managing scratch tape pools, you must define the storage groups as tape storage groups. Refer to [\*z/OS DFSMSrmm Implementation and Customization Guide\*](#).

For more information about tape storage groups and tape libraries, refer to [\*z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries\*](#).

## 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 (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.

A *stripe* is the portion of a striped data set that resides on one volume. 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 physical sequential (PS) data sets is 59. For VSAM data sets, the maximum number of stripes is 16. Only sequential or VSAM data sets can be striped.

Figure 4 on page 30 shows the formula that is used to derive the optimum number of stripes that are allocated to the data set:

$$\text{Sustained data rate} / \text{Device transfer rate} = \text{Optimum number of stripes}$$

*Figure 4. 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 assign data sets to tape only if they 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 disk sublevel 1 (Db2 tables), disk sublevel 2 (file system), optical volumes for the storage of objects on optical media, tape volumes for the storage of objects on tape media, and cloud storage. Tape volumes can be defined as sublevel 1 or sublevel 2.

OAM utilizes the SMS Storage Class construct to control where in the OAM storage hierarchy an object should be placed. The storage class is used during application initiated invocations such as OSREQ STORE and OSREQ CHANGE as well as during OSMC class transition. The determination of where an object should be placed is based on the values for:

- Initial Access Response Seconds (IARS)
- Sustained Data Rate (SDR)
- OAM Sublevel (OSL)

This is currently when IARS=0 objects are directed to the "disk" level of the OAM storage hierarchy (DB2 if OSL=1 or file system if OSL=2), when IARS>0 and SDR>=3 objects are directed to the "tape" level of the OAM storage hierarchy (Tape sublevel 1 if OSL=1 or Tape sublevel 2 if OSL=2), and when IARS>0 and SDR<3 objects are directed to the "optical" level of the OAM storage hierarchy (which is only supported for a Classic configuration).

To support specification of the cloud level, the meaning of IARS=0 is expanded to include both the disk and cloud levels and a new value of 3 is supported for OSL when IARS=0 to indicate the cloud level.

OAM has a dependency on the ISMF and SMS components to modify the existing OAM Sublevel field to support 3 as a valid value when IARS=0. The following shows the Storage Class Define ISMF panel with 3 added as a valid value for OAM Sublevel.

**Note:** 3 is only valid when Initial Access Response Seconds is 0.

Similarly, if it has been determined that the primary copy of the object is to be stored on tape, then if the OSL parameter of the storage class equals 1, the primary copy of the object is stored on a tape sublevel 1 volume. If the OSL parameter of the storage class equals 2, the primary copy of the object is stored on a tape sublevel 2 volume.

Table 1 on page 32 provides a summary of how the IARS, SDR, and OSL values determine the initial placement of an object in the OAM object storage hierarchy.

Table 1. Summary of media selection for object storage			
IARS value	SDR value	OSL value	Where object is stored
0	N/A	1	disk sublevel 1 (Db2 tables)
0	N/A	2	disk sublevel 2 (file system)
0	N/A	3	cloud level
non-zero	<3	N/A	optical disk
non-zero	≥3	1	tape sublevel 1
non-zero	≥3	2	tape sublevel 2

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

**Before you begin:** You must allocate an SCDS.

Perform the following steps to define storage groups.

1. Select option 6, Storage Group, from the ISMF Primary Option Menu for storage administrators. This displays the Storage Group Application Selection panel.

2. Supply values on the Storage Group Application Selection panel:

### CDS Name

This is the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active'. This represents the currently active configuration, but you cannot define or alter the storage groups for the active configuration.

### Storage Group Name

This is the storage group name. ISMF primes the Storage Group Name 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.

### Storage Group Type

See “[Storage group types](#)” on page 33 for more information.

### Option

Select option 3, Define. This displays the panels for defining the storage group. These vary with the storage group type.

3. Complete the panels for defining the storage group. See [“Storage group types” on page 33](#) for more information.

You can leave any of the Storage Group Define panels at any time without saving the storage group by issuing the Cancel command.

The remainder of this topic provides the panel sequences in the Storage Group Application and explains the attribute values for each type of storage group.

## Storage group types

The valid values for Storage Group Type are:

### 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.

See [“Defining a VIO storage group” on page 34](#) for more information.

### Pool

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

See [“Defining a pool storage group” on page 35](#) for more information.

### 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 wanted 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.

See [“Defining a dummy storage group” on page 44](#) for more information.

### Copy Pool Backup

Copy pool backup storage groups contain the target volumes of fast replication backup requests. Ensure that the number of eligible target volumes in a copy pool backup storage group is sufficient to satisfy the needs of the number of backup versions that are specified in its associated copy pool.

To be eligible for fast replication backup, a target volume must:

- Have the same track format as the source volume.
- Be the same size as the source volume.
- For FlashCopy®:
  - Not be a primary or secondary volume in an XRC or PPRC volume pair.
  - Not be in a FlashCopy relationship at the time of the backup.
- For SnapShot, the volume must reside in the same RVA/SVA as the source volume.

Volumes that are associated with copy pool backup storage groups are for DFSMSHsm use. Do not use these volumes for SMS volume allocation, or the allocation fails.

See [“Defining a copy pool backup storage group” on page 45](#) for more information.

For information about using ISMF to define a copy pool, see [Chapter 12, “Defining copy pools,” on page 143](#).

For more information about FlashCopy and SnapShot, see [z/OS DFSMS Advanced Copy Services](#).

### **Object**

Object storage groups identify an object storage hierarchy.

You can define object storage groups with no optical libraries. If you want the backup copies of objects to be written to tape, specify SETOAM statements in the CBROAMxx member of PARMLIB.

See [“Defining an object storage group” on page 46](#) for more information.

### **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 that are written to tape, cloud, file system based on the CBROAMxx member of PARMLIB.

See [“Defining an object backup storage group” on page 48](#) for more information.

### **Tape**

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

See [“Defining a tape storage group” on page 51](#) for more information.

## **Defining a VIO storage group**

Perform the following steps to define a VIO storage group.

1. Specify VIO for storage group type on the Storage Group Application Selection panel, as explained in [“Defining storage group attributes” on page 32](#). This causes the VIO Storage Group Define panel to be displayed.

- 
2. Supply values on the VIO Storage Group Define panel. 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.

You specify:

#### **Description**

This is an optional field of 120 characters in which you can describe the VIO storage group.

#### **VIO Maxsize**

This is a data set maximum size for VIO. See [“Values for defining a VIO storage group” on page 35](#).

#### **VIO Unit**

This represents the generic DASD device type (for example, 3380 or 3390) that this storage group simulates.

#### **System/Sys Group Name and Status**

This is a storage group status for each system and system group.

See [“Values for defining a VIO storage group” on page 35](#) for more information.

- 
3. Use the END command to save the VIO storage group and return to the Storage Group Application Selection panel.

## Values for defining a VIO storage group

### VIO Maxsize

This 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.



**Attention:** 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 affecting 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.

## Defining a pool storage group

Perform the following steps to define a pool storage group.

1. Specify POOL for storage group type on the Storage Group Application Selection panel, as explained in [“Defining storage group attributes” on page 32](#). This causes the Pool Storage Group Define panel to be displayed.  

---
2. Supply values on 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. See [“Values for defining a pool storage group” on page 35](#) for information on the other values.  

---
3. Use the END command to save the storage group and return to the Storage Group Application Selection panel. If you want to define SMS storage group status, press Enter. See [“Defining pool storage group status” on page 42](#) for more information.

## Values for defining a pool storage group

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, which is 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 specify the following attributes on the Pool Storage Group Define panel:

### Description

Is an optional field of 120 characters with which you can describe the pool storage group.

### 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**

(yes) specifies that data sets are eligible for primary space management and that DFSMSHsm performs automatic space management if a processing window is defined to DFSMSHsm. If the

DFSMSHsm system setting SETSYS INTERVALMIGRATION is specified, the data sets are eligible for interval migration. If the DFSMSHsm system setting SETSYS ONDEMANDMIGRATION(Y) is specified, the data sets are eligible for on-demand migration and are not eligible for interval migration, even if SETSYS INTERVALMIGRATION is specified.

**N**

(no) specifies that data sets are not eligible for automatic migration.

**I**

(interval) specifies that data sets are eligible for primary space management and forces DFSMSHsm to perform automatic interval migration independent of the DFSMSHsm system setting for SETSYS INTERVALMIGRATION. Interval migration is performed on volumes that are at or above the high threshold only. Interval migration is performed hourly. If a low threshold value of 0 is specified, DFSMSHsm migrates all eligible data sets in the selected storage group. DFSMSHsm on-demand migration is not performed. This value is most useful for storage groups used with tape mount management.

**P**

(primary) specifies that data sets are eligible for primary space management. Interval migration and on-demand migration are not performed regardless of the DFSMSHsm system setting for SETSYS INTERVALMIGRATION and SETSYS ONDEMANDMIGRATION.

**Note:** 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, you must specify Auto Migrate as Y, I or P.

**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, because 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 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.

Figure 2 on page 8 provides a visual representation of the system grouping concept.

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. Overflow is a required field, which ISMF primes with the value N (No). Y (Yes) specifies that the storage group is an overflow storage group.

To make overflow storage groups eligible for allocation, you must assign designated overflow storage groups in your storage group ACS routine.

If all volumes in a non-overflow storage group are so full that the current allocation request will push them over high threshold, while volumes in the overflow storage group are not so full, then the new data set will be allocated on a volume in the overflow storage group. The assumption is that all other attributes of the non-overflow storage group and overflow storage group and the volumes in those storage groups are the same.

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 non-overflow storage group, specified volume counts might result in overflow volumes being preferred over volumes in the non-overflow 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.

### 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 a combination, 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 101.

### Note:

1. **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.
2. Because an extend storage group is a pool storage group, the SMS status also applies to the extend storage group.



3. You can use the exit IGGPRE00\_EXIT to request SMS to switch to the extend storage group without attempting the extend request on any volumes in the primary storage group(s). When the exit encounters the volumes from the associated extend storage group, the exit is still able to decide whether to accept the extent. See the manual [DADSM Pre- and Post processing Dynamic Exits \(IGGP00\\_EXIT, IGGPOST0\\_EXIT\)](#) in *z/OS DFSMS Installation Exits* for more details on the exit IGGPRE00\_EXIT.

### **Copy Pool Backup SG Name**

Specifies the name of the copy pool backup storage group that contains eligible volumes for fast replication backup versions.

You can specify the name of a copy pool backup storage group that is shared by more than one pool storage group.

Copy Pool Backup SG Name is an optional field of one-to-eight alphanumeric or national characters, or a combination, the first of which must be alphabetic. ISMF primes the field with the default value of a blank name.

### **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 first define their names and parameters using DFSMSHsm. Then you can identify the dump class names to SMS using this panel.

Dump Class is an optional field. ISMF primes the field with the default value "blank."

### **DEFINE SMS Storage Group Status**

Specifies whether you want to change the status of the pool storage group with respect to a given system in the SMS complex. Initially, the status is set to ENABLE for all. The values are

**Y**

Change a status. When you complete the current panel, you will see the SMS Storage Group Status Define panel. For more information, see ["Defining pool storage group status"](#) on page 42.

**N**

Leave all the status fields as ENABLE. This is the default.

### **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 thresholds by looking at the primary space allocation of each data set 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. When a volume reaches or exceeds the high threshold, SMS issues an ENF72 signal, regardless of the Auto Migrate setting.

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 midpoint of the high and low thresholds. Because 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. Because SMS needs the value for allocation purposes, the High Threshold field is a required field with a primed value of 85.

For a track-managed space of an extended address volume, the allocation and migration threshold specifies the threshold percentage of space allocation that triggers or stops migration of data sets from volumes in this storage group during interval migration. Valid values for high threshold are 1-100. Valid values for low threshold are 0-99. These fields are primed with default values and are ignored by SMS and DFSMSHsm for other than extended address volumes. For the default values, refer to the ISMF help. If you alter an existing storage group to add extended address volume values, the specified bits are set to ON, when you specify the BreakPointValue and the Track Allocation Thresholds. If the specified indicators are not ON, SMS uses the existing threshold values, which represent threshold values for the entire volume, to represent threshold values for the track-managed space.

For more information about specifying allocation and migration thresholds, see [\*z/OS DFSMSHsm Storage Administration\*](#).

For more information about selecting appropriate threshold levels, see [\*z/OS DFSMS Implementing System-Managed Storage\*](#).

### **Total Space Alert Threshold %**

Specifies the minimum percentage (0-99) of total space usage that causes an alert message IGD400I to be issued.

**Note:** The following description of the storage group space alert message issuance applies to both the Total Space Alert Threshold and the Track-Managed Space Alert Threshold attributes.

DFSMS calculates the space usage of a pool storage group when a space change occurs on an online and ENABLE/QUINEW status volume in the pool storage group or when an enabled or QUINEW volume in the pool storage group is varied online or offline.

Rather than issuing the new alert message whenever DFSMS recalculates the space usage of a pool storage group, DFSMS will issue alert messages based on space usage % intervals. If the space usage of a pool storage group exceeds the Space Alert Threshold %, then the alert messages will be issued once at every 10% intervals from the threshold value (and at 5% intervals when the interval is greater than or equal to 85%) or when the space usage is 100%. See the following example.

When the space usage of a pool storage group decreases, alert messages will be reissued at every interval-and-a-half. This additional half-interval buffer will prevent alert messages from being over-issued due to the "ping-pong effect" in which the usage rate of a storage group bounces around a threshold value. This effect can occur when small data sets are being allocated and deleted repeatedly on volumes of a pool storage group. However, when the interval range is within 0-10%, there is no additional buffer to allow ease of removing the IGD400I and IGD401I message at low usage %.

For example, the Space Alert Threshold % attribute is set to 55%. The intervals at which an alert message is issued will be as follows: 55%---65%---75%---85%---90%---95%---100%. Note that each interval length is 10%, up until it reaches 85%, then the interval length decreases to 5%. This allows more frequent alert messages to be issued as the space usage of a storage group approaches 100%.

Scenarios to consider:

- If SMS calculates that the space usage of a pool storage group increased from 67% to 74%, then no new alert message will be issued. The current highlighted alert message is still outstanding and sufficient to monitor the storage. If the space usage exceeds 75%, increasing further from 74% to 77%, then the current highlighted alert message will be removed and the new alert message with the new space usage % will be issued.
- If DFSMS calculates that the space usage of a pool storage group decreased from 77%, the space usage must be less than or equal to 70% (an interval-and-a-half, since here an interval is 10% and the half is an additional 5%) to issue an alert message.

- Similarly, if DFSMS calculates that the space usage of a pool storage group decreases from 96%, the space usage must be less than or equal to 93% for an alert message to be issued (interval is 5%, so half is 2%).

#### **Track-Managed Space Alert Threshold %**

Specifies the minimum percentage (0-99) of track-managed space usage that causes an alert message IGD401I to be issued.

Messages are issued and deleted according to the same rules that are used with Total Space Alert Threshold %. Refer to the description of that attribute for details.

#### **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.

#### **BreakPointValue**

Specifies the disk space request (primary or secondary), expressed in the number of cylinders (0-65520), where the system should prefer the cylinder-managed space on an Extended Address Volume (EAV).

BreakPointValue applies only to data sets that are EAS (Extended Addressing Space)-eligible. Data sets that are not EAS-eligible must reside in the track-managed region of a volume.

When the requested disk space is greater than or equal to the value for BreakPointValue, the system prefers to use the cylinder-managed space for that extent.

- For all data set types, if not enough cylinder-managed space is available, the system uses track-managed space or both cylinder-managed space and track-managed space.
  - For non-VSAM data sets and VSAM data sets (excluding KSDS), when volumes in the volume list do not have enough cylinder-managed space and the primary storage groups have sufficient track-managed space, volumes that are enabled will be marked as primary. This will prevent a data set from being allocated in an overflow storage group.
- When the requested disk space is less than the value for BreakPointValue, the system prefers to use track-managed space for that extent.
  - For all data set types, if not enough track-managed space is available, the system uses cylinder-managed space or both track-managed space and cylinder-managed space.
  - For non-VSAM data sets and VSAM data sets (excluding KSDS), when volumes in the volume list do not have enough track-managed space and the primary storage groups have sufficient cylinder-managed space, volumes that are enabled will be marked as primary. This will prevent a data set from being allocated in an overflow storage group.

BreakPointValue is optional. If it is not specified for the storage group, the system uses the default value specified in the IGDSMSxx member of SYS1.PARMLIB, or a value of 10 cylinders if no BreakPointValue is specified in IGDSMSxx.

#### **Processing Priority**

Specifies the processing priority of the storage group. The value is in the range from 1 (lowest priority) to 100 (highest priority). It is used during DFSMSshm space management. The default is 50.

### **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 SMS from selecting the volume when a data set needs to extend to a new volume. For a data set that exists on a volume marked as DISNEW, extends on the same volume will be allowed.

#### **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 for Storage Groups. You can, however, change it for volumes.

Use the VARY command to change the status of a volume from NOTCON to another 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. If you use the VARY command to change the status of a volume the change is made in the ACDS and not the SCDS, so the status will change to whatever is defined in the SCDS if an SCDS is subsequently activated.

#### **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 102](#).

#### **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 (specific to JES3).

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, because 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 JES2, Quiesce New will not prevent the system from allocating new data sets. Under both JES2 and JES3, the data sets can extend to a new volume whose volume or storage group status is QUINEW. 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 102](#).

**Note:** Because an extend storage group is a pool storage group, the SMS status also applies to the extend storage group.

## Defining pool storage group status

**Before you begin:** You must specify Y for DEFINE SMS Storage Group Status on the Pool Storage Group Define panel, as described in [“Defining a pool storage group” on page 35](#).

Perform the following steps to define pool storage group status.

1. Supply values on the SMS Storage Group Status panel,

### **System/Sys Group Name**

Lists the system or system group names you defined in the base configuration.

### **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 40](#).

- 
2. Use the END command to save your values and return to the Pool Storage Group Define panel.

- 
3. Use the END command again (on the Pool Storage Group Define panel) to save the pool storage group. This returns you to the Storage Group Application Selection panel.

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. For more information, see [“Adding volumes and defining the SMS volume status” on page 42](#).

## Adding volumes and defining the SMS volume status

**Before you begin:** You must have displayed the Storage Group Application Selection panel (select option 6, Storage Group, from the ISMF Primary Option Menu).

Perform the following steps to add volumes and define the SMS volume status.

1. On the Storage Group Application Selection panel, select option 5, Volume. This displays the Storage Group Volume Selection panel.

- 
2. Supply values on the Storage Group Volume Selection panel:

- Select option 2, Define.
- 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. For an example, see [Figure 5 on page 43](#).

- 
3. Supply values on the SMS Volume Status Define panel. For more information on supplying values, see [“Values on the SMS volume status define panel” on page 43](#).

- 
4. Issue the END command to return to the Storage Group Application Selection panel. You can define additional DASD volumes to the pool storage group.

```

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:
-----
SYSTEM1  ==>  ENABLE      SYSTEM2  ==>  ENABLE      NOTCON, ENABLE,
SYSTEM3  ==>  ENABLE      SYSTEM4  ==>  ENABLE      DISALL, DISNEW,
SYSTEM5  ==>  ENABLE      SYSTEM6  ==>  ENABLE      QUIALL, QUINEW )
*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 5. Defining Volume System Status

**Values on the SMS volume status define panel**

On 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.

**Volume Serial Numbers**

Lists the volumes defined to the storage group.

**System/Sys Group name**

Lists the systems you defined in the base configuration.

**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 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 40 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

**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.

**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.

## Defining a dummy storage group

Perform the following steps to define a dummy storage group.

1. Specify DUMMY for storage group type on the Storage Group Application Selection panel, as explained in [“Defining storage group attributes”](#) on page 32. This causes the Dummy Storage Group Define panel to be displayed.

- 
2. Supply a description on the Dummy Storage Group Define panel, if desired. The description is optional. It can be up to 120 characters. 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.

- 
3. Use the END command to save the newly defined dummy storage group and return to the Storage Group Application Selection panel.

- 
4. Select option 5, Volume, on the Storage Group Application Selection panel to display the Storage Group Volume Selection panel, where you can add volume serial numbers to the storage group. (Initially, the storage group contains no volume serial numbers.)

ISMF primes the other fields on the Storage Group Application Selection panel with the most recently specified values. You do not need to specify a Storage Group Type.

- 
5. Supply values on the Storage Group Volume Selection panel to define volume serial numbers to the dummy storage group. For more information, see [“Defining DASD volumes to a dummy storage group”](#) on page 44.

## Defining DASD volumes to a dummy storage group

To add DASD volumes to the dummy storage group, supply values on the Storage Group Volume Selection panel:

1. Select option 2, Define.
2. 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 Type to include hexadecimal values in your range or an A under Type to include alphabetic values in your range. If you leave the Type field blank, only decimal values will be used in your range.

### Note:

1. The number of characters in the FROM and TO fields must be the same.
2. If TYPE is A, only one character is allowed in the FROM and TO fields

3. The value of the FROM field must be less than or equal to the value of the TO field.

For example, in Figure 6 on page 45, volume numbers SYS001 through SYS077 are specified on the first volume specification line (only decimal numbers will be used in the range). 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 NVOL through PVOL are specified on the fourth line, which specifies alphabetic values. You can add as many as a hundred volume serial numbers at a time to a pool storage group this way.

```
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 & Copy Pool Backup only)
     2. Define  - Add Volumes to Volume Serial Number List
     3. Alter   - Alter Volume Statuses (Pool & Copy Pool Backup only)
     4. Delete  - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
  Prefix  From    To    Suffix  Type
=====
===> SYS      001    077      -
===> DFPIP1
===> SYS      25     30     A       X
===>          N     P     VOL     A

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

Figure 6. Defining Dummy Storage Group Volume Serial Numbers

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 copy pool backup storage group

Perform the following steps to define a copy pool backup storage group.

1. Specify COPY POOL BACKUP for storage group type on the Storage Group Application Selection panel, as explained in [“Defining storage group attributes”](#) on page 32. This causes the Copy Pool Backup Storage Group Define panel to be displayed.

2. Supply a description on the Copy Pool Backup Storage Group Define panel, if desired. The description can be up to 120 characters. 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.

3. You can indicate the name for the copy pool backup storage group on the associated source pool storage group. For more information, see [“Defining a pool storage group”](#) on page 35.



## Defining an object storage group

**Before you begin:** 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.

Perform the following steps to define an object storage group.

1. Specify OBJECT for storage group type on the Storage Group Application Selection panel, as explained in [“Defining storage group attributes”](#) on page 32. This causes the Object Storage Group Define panel to be displayed.

- 
2. Supply values on the Object 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. For information on the other fields, see [“Attributes of the object storage group”](#) on page 46.

Use the DOWN command to advance to the next page as needed.

- 
3. Use END to save the definition, or press Enter if defining values for status. See [“Defining object or object backup storage group status”](#) on page 50 for more information.

## Attributes of the object storage group

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

### Description

This is an optional description of the object storage group. It can be up to 120 characters.

### Qualifier

This identifies a Db2 database on DASD that 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 that 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 identifies the beginning of a window of time. Use this field in association with Cycle End Time to specify a specific window of time in which object processing can be started for this storage group. Specify a value 0 - 23, or NONE. A value 0 - 23 represents an hour of the day.

For example, specify 00 for midnight; 01 for 1 AM; or 23 for 11 PM

Specify NONE if you do not want automatic processing for the storage group.

The value for Cycle Start Time must be different from the value for Cycle End Time. When you specify NONE, the value for Cycle End Time must be blank.

**Recommendation:** If you are running in an OAMplex and specify a cycle start time, specify an OSMC processing system name. Otherwise, object processing for that object storage group starts on all systems in the OAMplex at cycle start time.

### Cycle End Time

Identifies the end of a window of time when object processing can be started for this storage group.

The function of the cycle end time depends on which cycle window mode is in effect when OAM initializes. The cycle window mode can be specified by using the CYCLEWINDOW keyword for the SETOSMC statement in the CBROAMxx parmlib member (refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*). In Start Only mode, the cycle end time identifies the end of a window of time when object processing can be started for this storage group. In Start Stop mode, when the cycle end time has been reached, processing is the same as if



a STOP command had been issued for that storage group. No new work is scheduled and all work in progress is allowed to complete.

Specify a value 0 - 23, or blank. A value 0 - 23 represents an hour of the day, and is required when a value 0 - 23 was specified for Cycle Start Time.

For example, specify an hour of the day as 00 for midnight; 01 for 1 AM; or 23 for 11 PM

The value for Cycle End Time must be different from the 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. This is the one that 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 that you specify are used to process write requests to the storage group. Names must be valid real or pseudo optical library names that are defined in the SCDS.

### **Volume Full Threshold**

This is the number of free KB that triggers 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 as being 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 0 - 9999.

### **Drive Start Threshold**

This is the maximum number of object write requests that are outstanding for an optical drive in this storage group. When the number of object write requests to this storage group, which is 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 0 - 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.

### **DEFINE SMS Storage Group Status**

Indicates whether you want to change the storage group status. The values are:

**Y**

Change the status. When you complete the current panel, you will see the SMS Storage Group Status Define panel. For more information, see [“Defining object or object backup storage group status” on page 50](#). If you are defining a storage group, Y is the default value.

**N**

Do not change the status. If you are altering a storage group, N is the default value.

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.

### **OAM DB2 ID**

This is a 1- to 4-character SSID or Group Attachment Name that identifies a Db2 subsystem that an OAM Object instance in a multiple OAM configuration is associated with. Therefore, it also identifies which OAM Object instance in a multiple OAM configuration this object storage group is associated with.

### **OAM Deletion Protection**

This field specifies the deletion-protection mode for all objects in this object storage group. When deletion-protection is enabled, objects in this object storage group cannot be deleted before their expiration date. Deletion-protection does not restrict any changes to an object's expiration date.

You must have also specified DP=P in the IEFSSNxx PARMLIB member. This field is ignored when no value is specified for DP in PARMLIB member IEFSSNxx, and when DP=A or DP=N is specified.

### **OAM Retention Protection**

This field indicates whether new objects that are stored into an object storage group are flagged as retention-protected for the entire life of the objects. A retention-protected object cannot be deleted before its expiration date, and its expiration date can never move to an earlier date.

OAM Retention Protection is used by OAM to set the retention-protection mode for an object at the time that the object is stored. If OAM Retention Protection is enabled at the time that the object is stored, then the object is in retention-protection mode for the life of the object. Setting OAM Retention Protection to disabled affects the retention-protection mode of subsequent objects that are stored into this object storage group, but does not impact the retention-protection mode of objects that were previously stored.

## **Defining an object backup storage group**

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.

Perform the following steps to define an object backup storage group.

1. Specify OBJECT BACKUP for storage group type on the Storage Group Application Selection panel, as explained in [“Defining storage group attributes” on page 32](#). This causes the Object Backup Storage Group Define panel to be displayed.
2. Supply values on the Object Backup 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. For information on the other fields, see [“Attributes of the object backup storage group” on page 48](#).

Press Enter to verify the values and display the SMS Storage Group Status Define panel.

3. Supply values on the SMS Storage Group Status Define panel. See [“Defining object or object backup storage group status” on page 50](#) for more information. 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.
4. Use the END command to save the newly defined object storage group and return to the Storage Group Application Selection panel.

## **Attributes of the object backup storage group**

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

### **Description**

This is an optional value of up to 120 characters that describes the object backup storage group.

### **Cycle Start Time**

This identifies the beginning of a window of time. Use this field in association with Cycle End Time to specify a specific window of time in which object processing can be started for this object backup storage group.

Specify a value from 0 - 23, or NONE. A value from 0 - 23 represents an hour of the day. For example, specify 00 for midnight; 01 for 1 a.m.; or 23 for 11 p.m. Specify NONE if you do not want automatic processing for the storage group.

The value for Cycle Start Time must be different from the value for Cycle End Time. When you specify NONE, the value for Cycle End Time must be blank.

**Recommendation:** If you are running in an OAMplex and specify a cycle start time, specify an OSMC processing system name. Otherwise, object processing for that object backup storage group will start on all systems in the OAMplex at cycle start time.

### Cycle End Time

Identifies the end of a window of time when object processing can be started for this object backup storage group.

The function of the cycle end time depends on which cycle window mode is in effect when OAM initializes. The cycle window mode can be specified by using the CYCLEWINDOW keyword for the SETOSMC statement in the CBROAMxx parmlib member (refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*). In Start Only mode, the cycle end time identifies the end of a window of time when object processing can be started for this storage group. In Start Stop mode, when the cycle end time has been reached, processing will be the same as if a STOP command had been issued for that storage group. No new work will be scheduled, all work in progress will be allowed to complete.

Specify a value from 0 - 23, or blank. A value 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. For example, specify an hour of the day as 00 for midnight; 01 for 1 a.m.; or 23 for 11 p.m.

The value for Cycle End Time must be different from the 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 backup storage group. This prevents multiple systems from trying to process the same storage group at the same time, especially when the Cycle Start Time parameter is used for automatically starting storage group processing.

### Library Names

Specifies 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

Specifies 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

Specifies 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 you specify a library name.

### Volume Full at Write Error

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 you specify 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 you specify a library name.

#### **DEFINE SMS Storage Group Status**

Indicates whether you want to change the storage group status. This is a required field. The values are:

**Y**

(Yes). If you are defining a storage group, the default value is Y and cannot be changed to N. If you specify Y, you will see a panel for changing the storage group status next. See [“Defining object or object backup storage group status” on page 50](#) for more information.

**N**

(No). If you are altering a storage group, the default value is N and can be changed to Y or N.

Object tape parameters for object and object backup storage groups are specified with SETOAM statements in the CBROAMxx 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**

**Before you begin:** You must define an object storage group or object backup storage group, as described in [“Defining an object storage group” on page 46](#) and [“Defining an object backup storage group” on page 48](#). After you have supplied all required values on the Object Storage Group Define panel or the Object Backup Storage Group Define panel, the SMS Storage Group Status Define panel is displayed. 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.

Perform the following steps to define object or object backup storage group status.

1. Supply values on the SMS Storage Group Status Define panel. 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. For more information, see [“Attributes for object or object backup storage group status” on page 50](#).

- 
2. Use the END command to save your values and return to the Object Storage Group or Object Backup Storage Group Define panel.

- 
3. Use the END command again, on the Object Storage Group or Object Backup Storage Group Define panel, to complete and save the definition of the storage group.

## **Attributes for object or object backup storage group status**

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.

## 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 entry is initially defined in the Db2 Collection Name Table. The ACS routines provide storage class and management class names that are used as the default storage class and management class assignments. Both are recorded in the Db2 Collection Name Table entry and the storage group assignment for the collection. OAM uses the storage group assignment to identify the Db2 tables that are used for storage of the object directory information, for storage of objects in Db2 tables, and to select the file system, optical volume, tape volume, or cloud container that is used to store objects that belong to the collection.

## Defining a tape storage group

Perform the following steps to define a tape storage group.

1. Specify TAPE for storage group type on the Storage Group Application Selection panel, as explained in [“Defining storage group attributes” on page 32](#). This causes the Tape Storage Group Define panel to be displayed.

---

2. Supply values on the Tape 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. For information on other fields, see [“Attributes for tape storage groups” on page 51](#).

---

3. If the value for DEFINE SMS Storage Group Status is Y, press Enter to view the SMS Storage Group Status Define panel. See [“Defining tape storage group status” on page 52](#) for more information.

## Attributes for tape storage groups

**Description**

This is an optional value of up to 120 characters that describes the tape storage group.

**Library Names**

Specifies 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.

**DEFINE SMS Storage Group Status**

This 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. The values are:

## Y

If you specify Y (Yes), the SMS Storage Group Status Define panel is displayed when you press Enter. See [“Defining tape storage group status” on page 52](#) for more information.

## N

If you specify N (No), the storage group defaults to enable for each one of the systems in the complex. N is the default.

## Defining tape storage group status

**Before you begin:** If you specified Y for the Define SMS Storage Group Status attribute on the Tape Storage Group Define panel, as described in [“Defining a tape storage group” on page 51](#), you see the SMS Storage Group Status Define panel, where you can define the status of the storage group to each system in the SMS complex.

Perform the following steps to define tape storage group status.

1. Supply values on the SMS Storage Group Status Define panel. 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.

### SMS SG Status

This 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 40](#) for an explanation of these relationships.

- 
2. Use the END command to save your values and return to the Tape Storage Group Define panel.

- 
3. Use the END command again, on the Tape Storage Group Define panel, to complete and save the definition of the 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 components” on page 208](#).

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

Perform the following steps to list volumes for a pool, dummy, copy pool backup, object, object backup storage group or tape storage group. Volumes are not associated with VIO storage groups.

1. Select option 6, Storage Group, on the ISMF Primary Option Menu for storage administrators. This displays the Storage Group Application Selection panel.  
-----
2. Select option 1, List, on the Storage Group Application Selection panel.  
-----
3. Optionally, type / next to one or more list options, which allow you to specify view or sort criteria, and to see space information in gigabytes (GB).  
-----
4. Type LISTVOL in the Line Operator column next to a storage group. This displays volume information, such as volume serial numbers, for the storage group. The LISTVOL line operator without any parameters lists all volumes in a storage group. The list displayed depends on the storage group type. See [Table 2 on page 53](#) for more information.

Table 2. Volumes Listed in a Specified Storage Group

Storage Group Type	Volume List Shown
<b>POOL or DUMMY</b>	DASD Volume List
<b>COPY POOL BACKUP</b>	DASD Volume List
<b>OBJECT or OBJECT BACKUP</b>	Mountable Optical Volume List
<b>TAPE</b>	Mountable Tape Volume List

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. The STATUS line operator can be abbreviated to ST.

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

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





---

## Chapter 5. Defining reserve storage pools

This topic describes reserve storage pools.

### Understanding reserve storage pools

---

Reserve storage pools are groups of volumes that have been formatted for future use. Reserve storage pools can make it easier for you to manage your defined but unused volumes. You give each reserve storage pool a name. You can have any number of reserve storage pools.

Using reserve storage pools can make it easier to manage your defined but unused volumes.

To belong to a reserve storage pool, a volume must have been initialized as reserved and have an owner ID beginning with IBMRSP followed by the name of a reserve storage pool. It then has a volume serial number but cannot be brought online. It remains offline until it is re-initialized as a non-reserved volume.

### Assigning volumes to a reserve storage pool

---

To assign a volume to a reserve storage pool, you initialize the volume using the ICKDSF INIT command with these parameters:

**RESERVED**

Initializes the volume as reserved, which specifies that it cannot be brought online.

**OWNERID**

Specifies the name of the reserve storage pool, in the form IBMRSP`rspname` where `rspname` is the name of the reserved storage pool (up to 8 characters). OWNERID is optional.

A volume that has been initialized as reserved cannot be brought online. Before it can be brought online, it must be re-initialized without the RESERVED keyword. For more information, refer to [INIT command—CKD](#) in *Device Support Facilities (ICKDSF) User's Guide and Reference*.

A volume that has been initialized as reserved is not to be confused with a device for which a system issues a hardware RESERVE, either through the RESERVE or ISGENQ macros, to ensure successful serialization of shared DASD. For more information, refer to [z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU](#) or [z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG](#).



---

# Chapter 6. Defining cloud network connections

z/OS DFSMS simplifies the management of object storage cloud definitions by introducing a network connection construct. This topic describes DFSMS network connection construct and explains how you can define them with the ISMF network connection application.

The network connection construct begins with a cloud network connection name, which specifies the name of the cloud that DFSMS and the DS8000® storage subsystem is to connect to. The maximum name length is 30 alphanumeric or national (\$, @, #) characters, or any combination. The first character cannot be numeric.

After you define a cloud, you can use DFSMSdss and DFSMSshm commands to archive data sets to an object storage cloud.

Reference information:

- For information about using DFSMSshm commands to perform migrate, recall, list, audit, report and delete functions, see [z/OS DFSMSshm Storage Administration](#).
- For information about using DFSMSdss commands to perform dump and restores, see [z/OS DFSMSdss Storage Administration](#).

---

## Understanding clouds

The network connection construct begins with a *cloud network connection name*, which specifies the name of the cloud that DFSMS and the DS8000 storage subsystem is to connect to. The maximum name length is 30 alphanumeric or national (\$, @, #) characters, or any combination. The first character cannot be numeric.

After you define a cloud, you can use DFSMSdss and DFSMSshm commands to archive data sets to an object storage cloud.

Reference information:

- For information about using DFSMSshm commands to perform migrate, recall, list, audit, report and delete functions, see [z/OS DFSMSshm Storage Administration](#).
- For information about using DFSMSdss commands to perform dump and restores, see [z/OS DFSMSdss Storage Administration](#).

---

## Planning a cloud

- Enable cloud functions on your DS8000, as specified in the DS8000 information.
- Obtain the following information from your cloud provider or administrator:

Name	Description
<b>Provider</b>	Two choices are available: <ul style="list-style-type: none"><li>– SWIFT, if your cloud is an Openstack-based Swift object storage cloud that uses the built-in Tempauth authorization system.</li><li>– SWIFT-KEYSTONE, if your cloud is an Openstack-based Swift object storage cloud that uses the Keystone Identity service.</li></ul>
<b>Identity</b>	A concatenation of the tenant name and a username in the form <i>tenant_name:user_name</i> .
<b>Endpoint</b>	The Uniform Resource Identifier (URL) that DFSMS should use for authentication with your object storage cloud.
<b>Port number</b>	The port on which DFSMS is to communicate with the endpoint.

Name	Description
<b>SSL version</b>	The lowest acceptable level of SSL/TLS that can be used when a secure connection is being established.

## Defining a cloud

You can use ISMF to define, alter, list, or display network connection construct. This topic describes the ISMF panels that you use to define network connection attributes and outlines the steps to define a cloud network connection.

Using ISMF panels, define a network connection construct that corresponds in name to a cloud that is defined in the DS8000 configuration.

- On the **ISMF Primary Option Menu** panel, select **S** to specify network connection attributes.
- On the **Cloud Application Selection** panel, specify the CDS name and the cloud network connection name, and select option 3 to Define a cloud.
- On the **Network Connection Define/Alter** panel, fill in the following fields, with information obtained from your cloud provider or administrator:

Name	Description
<b>Cloud connection network name</b>	The name of the cloud in the selected CDS.
<b>Provider</b>	The provider of the cloud (for example, SWIFT or SWIFT-Keystone).
<b>Identity (credentials)</b>	The credentials used when authenticating with the cloud. This is a concatenation of the tenant name and a username for that tenant in the form <i>tenant_name:user_name</i> .
<b>Endpoint</b>	The Uniform Resource Identifier (URL) that DFSMS should use when authenticating with the cloud. When SWIFT-KEYSTONE is the provider, the endpoint must contain the version number of the identity API to use. Only the version 2 API is supported. For example, if the endpoint is <code>https://dallas.ibm.com</code> the endpoint should be <code>https://dallas.ibm.com/v2.0</code> .
<b>Port number</b>	The remote port number to which to connect instead of the default HTTP or HTTPS port. Maximum length: five characters, valid values: 0 to 65535.
<b>SSL version</b>	The lowest SSL version acceptable to use when making HTTP requests. Maximum length: eight characters, valid values: TLSV12, TLSV11, TLSV1, SSLV3, blank.
<b>SSL key</b>	The name of the keystore to be used (required when SSL version is not blank). The value can be either a SAF key ring name in the form of <code>userid/keyring</code> , or a PKCS #11 token in the form of <code>*TOKEN*/token_name</code> .

## Steps for defining a DS8000 as an object proxy

Using ISMF panels, define a network connection construct that corresponds in name to a cloud that is defined in the DS8000 configuration.

- On the **ISMF Primary Option Menu** panel, select 'S' to specify network connection attributes.
- On the **Cloud Application Selection** panel, specify the CDS name and the cloud network connection name, and select option 3 to Define a cloud.
- On the **Network Connection Define/Alter** panel, fill in the following fields, with information obtained from your cloud provider or administrator:

Name	Description
<b>Cloud connection network name</b>	The name of the cloud in the selected CDS.
<b>Provider</b>	Swift.
<b>Identity (credentials)</b>	The name of a user that is defined to the DS8000 Hardware Management Console (HMC) that is to be used specifically for proxy operations.
<b>Endpoint</b>	The Uniform Resource Identifier (URL) of the DS8000 HMC that will be acting as the proxy server. HMC connections require https communication.
<b>Port number</b>	The remote port number to which to connect instead of the default HTTP or HTTPS port. Maximum length: five characters, valid values: 0 to 65535.
<b>SSL version</b>	The lowest SSL version acceptable to use when making HTTP requests. Maximum length: eight characters, valid values: TLSV12, TLSV11, TLSV1, SSLV3, blank.
<b>SSL key</b>	The name of the keystore to be used (required when SSL version is not blank). The value can be either a SAF key ring name in the form of userid/keyring, or a PKCS #11 token in the form of *TOKEN*/token_name.

## Setting up digital certificates for the cloud

If your cloud provider uses SSL/TLS communication, then you must also obtain the necessary digital certificates and understand the type of SSL/TLS authentication the cloud performs. Complete the following steps to set up the digital certificate.

To set up for server authentication, follow these steps:

1. Add the cloud root CA certificate:
  - a. `RACDCERT CERTAUTH ADD(<dataset containing cloud root CA>) WITHLABEL('Cloud Root CA') TRUST`
2. Add the cloud intermediate CA certificate:
  - a. (Optional) `RACDCERT CERTAUTH ADD(<dataset containing cloud intermediate CA>) WITHLABEL('Cloud Intermediate CA') TRUST`

The intermediate CA certificate can be omitted if the clouds key ring contains the intermediate CA certificate.

If the cloud identity certificate was signed by a root CA then the intermediate CA is also not necessary.
3. In this case we can use a virtual keyring in our SMS network connection construct:
  - a. The SSL key in the SMS network connection construct would be a virtual key ring by the name of `*AUTH*/*`.

## Setting up mutual certification

1. Add a key ring for DFSMS to use:
  - a. Details: `RACDCERT ID(DFHSM) ADDRING(ring_name)`
2. Add the z/OS host identity certificate with the private key and its CA certificate:
  - a. `RACDCERT CERTAUTH ADD(<dataset containing X>) WITHLABEL('zOS host Root CA') TRUST`
  - b. `RACDCERT ID(DFHSM) ADD(<dataset containing C and private key>) WITHLABEL('zOS host cert') TRUST PASSWORD('...')`
3. Add the cloud root CA certificate:

- a. `RACDCERT CERTAUTH ADD(<dataset containing A>) WITHLABEL('Cloud Root CA') TRUST`
4. Connect certificates to the CLOUD key ring:
  - a. `RACDCERT ID(DFHSM) CONNECT(CERTAUTH LABEL('zOS host Root CA') ring(CLOUD))`
  - b. `RACDCERT ID(DFHSM) CONNECT(CERTAUTH LABEL('Cloud Root CA') ring(CLOUD))`
  - c. `RACDCERT ID(DFHSM) CONNECT(LABEL('zOS host cert') ring(CLOUD) DEFAULT)`
5. In this case we use a real key ring in our SMS network connection construct:
  - a. The SSL key in the SMS network connection construct would be a key ring by the name of DFHSM/ring\_name.

## Setting up digital certificates when using DS8000 as an object proxy

If your cloud provider uses SSL/TLS communication, then you must also obtain the necessary digital certificates and understand the type of SSL/TLS authentication the cloud performs. Complete the following steps to set up the digital certificate.

- 1.

## Defining additional clouds

You can copy existing network connection construct and modify them to create new network connection constructs by using the COPY line operator, which is explained in [“Copying SMS components” on page 208](#).

---

## Chapter 7. Defining network connections

z/OS DFSMS simplifies the management of resources that are accessed by using the hypertext transfer protocol/hypertext transfer protocol secure (HTTP and HTTPS) over Transmission Control Protocol/Internet Protocol (TCP/IP) by introducing a network connection construct. This topic describes the DFSMS network connection construct and explains how you can use the ISMF network connection application to define them.

---

### Understanding network connections

A *network connection* is a set of attributes and their values that are used to connect to a resource over HTTP or HTTPS. Defining a connection to a resource does not establish the connection. Applications that use the network connection are responsible for establishing and managing their own connection.

The *network connection construct* begins with a network connection name. The maximum length is 30 alphanumeric or national (\$, @, #) characters, or any combination. The first character cannot be numeric.

---

### Securing your network communication

z/OS DFSMS supports the transport layer security (TLS) protocol and uses SAF key rings to manage the digital certificates used during the TLS handshake. TLS is used to authenticate the server and to ensure that the cloud credentials are encrypted before they are transmitted over the IP network. For an overview of TLS, see [Writing and building a z/OS System SSL application in z/OS Cryptographic Services System SSL Programming](#).

**Restriction:** DFSMS supports server authentication using digital certificates. It does not support client or mutual authentication.

---

### Setting up digital certificates for the cloud

z/OS DFSMS supports the transport layer security (TLS) protocol and uses SAF key rings to manage the digital certificates used during the TLS handshake. TLS is used to verify the authenticate the server and to ensure the cloud credentials are encrypted before transmitting over the IP network. For an overview of TLS, see [Writing and building a z/OS System SSL application in z/OS Cryptographic Services System SSL Programming](#).

**Restriction:** DFSMS supports server authentication using digital certificates. It does not support either client or mutual authentication.

## Setting up certificates signed by a certificate authority

### About this task

To set up a certificate that was signed by a certificate authority, follow these steps:

### Procedure

1. Obtain the digital certificates from your object storage provider.
2. Upload the files to your z/OS host. This upload must be done in ASCII.
3. Add the certificates to RACF under CERTAUTH.

- a) Add the root CA certificate:

```
RACDCERT CERTAUTH ADD (<dataset containing cloud root CA>)  
WITHLABEL ('Cloud Root CA') TRUST
```

- b) Add the intermediate CA certificate:

```
RACDCERT CERTAUTH ADD (<dataset containing cloud intermediate CA>)
WITHLABEL ('Cloud Intermediate CA') TRUST
```

4. Authorize access to the virtual key ring for the DFSMSHsm address space (`dfhsm-region-userid`). Do this by administering a profile in the FACILITY class.

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
```

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(dfhsm-region-userid) ACCESS(READ)
```

- a) If the FACILITY class is not already active, activate and RACLIST it.

```
SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
```

- b) If the FACILITY class is already active and RACLISTed, refresh it.

```
SETROPTS RACLIST(FACILITY) REFRESH
```

You could use the RDATA LIB class but if you are using virtual key rings, using the FACILITY class is simpler.

5. When you configure ISMF, use the virtual key ring by specifying its fully qualified name for the KEYRING directive: `*AUTH*/*`

## Results

When you are done, you have set up the certificate that was signed by a certificate authority.

## Setting up self-signed certificates

### About this task

To set up a self-signed certificate, follow these steps:

### Procedure

1. To obtain the HMC certificate, issue the following **openssl** command from a machine with **openssl** installed:

```
openssl x509 -in <(openssl s_client -connect DS8000-HMC-
IP:8452 -prexit 2>/dev/null) -text -out certificate.pem
```

Substitute DS8000-HMC-IP with the IP address of the HMC the certificate is being obtained from. This command will download the certificate for the HMC and store it in a file named `certificate.pem`.

2. Upload the files to your z/OS host. This upload must be done in ASCII.
3. Add the certificates to RACF under SITE.

```
RACDCERT SITE ADD('DS8000.SITECERT') WITHLABEL('DS8000 SELF
SIGNED SITE') TRUST
```

4. Authorize access to the virtual key ring for the DFSMSHsm address space (`dfhsm-region-userid`). Do this by administering a profile in the FACILITY class.

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
```

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(dfhsm-region-userid) ACCESS(READ)
```

- a) If the FACILITY class is not already active, activate and RACLIST it.

```
SETROPTS CLASSACT(FACILITY) RACLIST(FACILITY)
```

- b) If the FACILITY class is already active and RACLISTed, refresh it.



You could use the RDATA LIB class but if you are using virtual key rings, using the FACILITY class is simpler.

## Results

When you are done, you have set up the self-signed certificate.

## Object storage network connections

A *network connection* is used to connect to an object storage cloud. The connection name specifies a name that DFSMS and the storage system use to coordinate data transfers.

After you define an object storage network connection, you can use DFSMSdss and DFSMSHsm commands to archive data sets to an object storage cloud.

- For information about using DFSMSHsm commands to perform migrate, recall, list, audit, report and delete functions, see *z/OS DFSMSHsm Storage Administration*.
- For information about using DFSMSdss commands to perform dumps and restores, see *z/OS DFSMSdss Storage Administration*.

## Planning network connections

Obtain the following information from the owner of the network resource you are attempting to establish a connection to.

- *Provider* is the type of resource and protocol that will be used when communicating with the network resource.
- *Identity* is the username and for some providers, the tenant name, that is to be used when connecting to a network resource. Local directory access protocol (LDAP) user names of the form `user@domain` and `domain/user` are supported.
- *Endpoint* is the Uniform Resource Identifier (URL) that DFSMS should use for authentication with your network resource.
- *Port number* is the port on which DFSMS is to communicate with the endpoint.
- *SSL version* is the lowest acceptable level of SSL/TLS that should be used when establishing a secure connection.

## Defining network connections attributes

You can use ISMF to define, alter, list, or display the network connection construct. This topic describes the ISMF panels that you use to define network connection attributes and outlines the steps for defining a cloud network connection.

## Defining connections to object storage clouds

Using ISMF panels, define a network connection construct that corresponds in name to a cloud that is defined in the DS8000 configuration.

## Defining connections to a TS7700 for DS8000 object store

### About this task

DFSMS does not communicate directly with a virtual tape server that is acting as an object store, but it can use a DS8000 HMC as a gateway. To configure DFSMS for a virtual tape server that is acting as an object store, perform these steps:

## Procedure

1. On the **ISMF Primary Option Menu** panel, select S to specify network connection attributes.
2. On the **Cloud Application Selection** panel, specify the CDS name and the cloud network connection name, and select option 3 to Define a cloud.
3. On the **Network Connection Define/Alter** panel, complete the following fields with information from your cloud provider or administrator:

Name	Description
Cloud network connection name	The name of the cloud in the selected CDS.
Provider	TAPE-OBJECT
Endpoint	<p>The Uniform Resource Identifier (URL) that DFSMS should use when authenticating with the object storage cloud.</p> <p>When configuring a connection to use a virtual tape server as an object store, the endpoint will point to a DS8000 HMC. This connection will be used for administrative commands and metadata transfers. If you have multiple DS8000s, specify one of them as the endpoint.</p>
Identity	The name of a user with monitor access to the DS8000 HMC. LDAP user names are supported for those HMCs that are set up to use LDAP for authentication.
Port	8452
SSL version	TLSV12
SSL key	<p>The SSL key is either:</p> <p><b>*AUTH*/*</b> When the storage systems HMC certificate was signed by a certificate authority.</p> <p><b>*SITE*/*</b> When the storage systems HMC certificate was not signed by a certificate authority. That is, it was self-signed.</p>

## Defining connections to an s3-based object storage cloud service provider

### About this task

DFSMS does not natively support object stores that expose an S3-compatible interface, but it can use a DS8000 HMC as a gateway. To configure DFSMS for an object store that exposes an S3-compatible API, perform the following steps:

## Procedure

1. On the **ISMF Primary Option Menu** panel, select S to specify network connection attributes.
2. On the **Cloud Application Selection** panel, specify the CDS name and the cloud network connection name, and select option 3 to Define a cloud.
3. On the **Network Connection Define/Alter** panel, complete the following fields with information from your cloud provider or administrator:

Name	Description
Cloud network connection name	The name of the cloud in the selected CDS.

Name	Description
Provider	SWIFT  Even though the storage system is connected to an object store that exposes an S3-compatible API, DFSMS communicates with the DS8000 HMC using a Swift based protocol.
Endpoint	The Uniform Resource Identifier (URL) that DFSMS should use when authenticating with the object storage cloud.  When configuring a connection to an object store that exposes an S3 compatible API the endpoint will not be the object store itself, but it will be a storage systems HMC. This connection will be used for administrative commands and metadata transfers. If you have multiple storage systems, specify one of them as the endpoint
Identity	The name of a user with monitor access to the storage systems HMC. LDAP usernames are supported for those HMCs setup to use LDAP for authentication.
Port	8452
SSL version	TLSV12
SSL key	The SSL key is either:  <b>*AUTH*/*</b> When the storage systems HMC certificate was signed by a certificate authority.  <b>*SITE*/*</b> When the storage systems HMC certificate was not signed by a certificate authority. That is, it was self-signed.

## Defining connections to an OpenStack Swift-based object storage cloud service provider

### About this task

DFSMS does natively support object stores that expose an OpenStack Swift compatible interface. To configure DFSMS for an object store that exposes an S3 compatible API, perform the following steps:

### Procedure

1. On the **ISMF Primary Option Menu** panel, select S to specify network connection attributes.
2. On the **Cloud Application Selection** panel, specify the CDS name and the cloud network connection name, and select option 3 to Define a cloud.
3. On the **Network Connection Define/Alter** panel, complete the following fields with information from your cloud provider or administrator:

Name	Description
Cloud network connection name	The name of the cloud in the selected CDS.
Provider	SWIFT  Even though the storage system is connected to an object store that exposes an S3-compatible API, DFSMS communicates with the DS8000 HMC using a Swift-based protocol.

Name	Description
Endpoint	<p>The Uniform Resource Identifier (URL) that DFSMS should use when authenticating with the object storage cloud.</p> <p>When configuring a connection to an object store that exposes an S3-compatible API, the endpoint will not be the object store itself, but it will be a storage systems HMC. This connection is used for administrative commands and metadata transfers. If you have multiple storage systems, specify one of them as the endpoint.</p>
Identity	The credentials used when authenticating with the cloud. This is a concatenation of the tenant name and a username for that tenant in the form <i>tenant_name:user_name</i> .
Port	The remote port number to connect instead of the default HTTP or HTTPS port. The maximum length is five characters and valid values are 0 to 65535.
SSL version	The lowest SSL version acceptable to use when making HTTP requests. The maximum length is eight characters and valid values are TLSV12, TLSV11, TLSV1, SSLV3, blank.
SSL key	<p>The SSL key is either:</p> <p><b>*AUTH*/*</b> When the object stores certificate was signed by a certificate authority.</p> <p><b>*SITE*/*</b> When the object stores certificate was not signed by a certificate authority. For example, if you are using the certificate that was defined when the DS8000 was delivered.</p>

## Defining additional networks

You can copy existing network connection constructs and modify them to create new network connection constructs by using the COPY line operator, which is explained in [“Copying SMS components” on page 208](#).

---

## Chapter 8. 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 topic 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, class transition 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, backup or class transition).

### 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 DB2 Collection Name Table 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.

### 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. An Auto Backup of 'Y' indicates to OAM that backup copy(s) are required for this object. A Backup Frequency of zero indicates that OAM is to schedule the first backup copy be written at the time the object is initially stored, whereas a non-zero value results in the first backup copy being written during the first storage management cycle after the object has been stored. If two backup copies are desired, specify a value of two or greater in the management class Number of Backup Versions attribute, and specify the SETOSMC statements in your CBROAMxx 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 expiration of objects. 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, class definition 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
- Class transition criteria
- Transition copy technique
- Serialization exit

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, migration, or transitions, 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

---

Perform the following steps to define management classes.

1. Select option 3, Management Class, from the ISMF Primary Option Menu for storage administrators. This displays the Management Class Application Selection panel.

- 
2. Supply values on the Management Class Application Selection panel:

### **CDS Name**

This is the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active', which represents the currently active configuration. You cannot define or alter the management classes for the active configuration.

### **Management Class Name**

This is the name of the management class. 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.

### **Option**

Select option 3, Define. This displays the Management Class Define panel.

You can use the DOWN command to display the second page, where you can select an attribute group to display first. Selecting an attribute group is only useful when performing a display, define, or alter function. You select an attribute group for processing ahead of other Management Class attributes by entering its corresponding number in the selection field. You can display panels containing the other management class attributes by paging up and down in the usual manner.

- 
3. Supply values on the Management Class Define panel. Use the DOWN command to move to the next page.

For more information, see:

- [“Defining management class expiration attributes” on page 70](#)
- [“Defining management class migration attributes” on page 73](#)
- [“Defining management class backup attributes” on page 77](#)
- [“Defining class transition attributes” on page 79](#)
- [“Defining aggregate backup attributes” on page 80](#)

You can leave any page of the Management Class Define panel without saving the management class by issuing the Cancel command.

## Defining management class expiration attributes

Page 1 of the Management Class Define panel 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. Supply these values:

### Description

Is an optional field of 120 characters you can use to describe the management class.

### Expiration Attributes

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.

#### Expire after Days Non-usage

Specifies how much time must elapse since last access before a data set or object becomes eligible for expiration. The default is NOLIMIT.

#### Expire after Date/Days

Specifies an absolute date or period after its allocation for a data set or object to become eligible for expiration. The default is NOLIMIT.

### Retention Limit

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 data 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 a retention limit value of either zero or lower than the valued specified in RETPD/EXPDT, 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.

### Retention Method

Specifies how new tape volumes are to be managed by DFSMSrmm. The retention method for a volume set is chosen when its first file is written.

The retention method is an attribute of the volume.

The EXPDT retention method allows expiration on a specific date and time or based on an event (catalog retention or retention for a specific number of days after last use).

The VRSEL retention method uses vital record specifications to implement retention and movement policies through which a retention date is calculated each time that VRSEL inventory management processing is run. DFSMSrmm retains a volume based on this retention date and on the volume expiration date.

- Length: 5 characters
- Type: Input
- Required: No

Valid values:

- EXPDT - Retention and expiration is managed based on expiration date, date of last reference, or catalog status.
- VRSEL - Policy management is to be managed by DFSMSrmm based on VRSEs using existing, supported combinations of data set name, jobname and the MC name.



- blank - value is not specified.

Default: blank

### **Volume Set Management Level**

Specifies how DFSMSrmm is to retain volumes or multivolume sets that are managed by the EXPDT retention method.

- Length: 9 characters
- Type: Input
- Required: No

Valid values:

- VOLUME - The expiration date is determined separately for each volume in the set. Unless defined differently, the expiration date is the maximum of all data set expiration dates on the volume. Each file on a volume can increment the volume expiration date. All files of a multivolume data set have the same expiration date.
- FIRST - The expiration date of the first file is used to set the expiration date of the volume or multivolume set. All volumes in a set will have the exact same expiration date and will be released to scratch in the same run of DFSMSrmm inventory management.
- SET - The expiration date is the maximum of all data set expiration dates of all the volumes in the set. All volumes in the set will have the exact same expiration date. Any file on any volume of the set can increment the volume expiration date. All files of a multivolume data set have the same expiration date.
- blank - value is not specified.

Default: blank

**Note:** Volume Set Management Level is not allowed to be specified when Retention Method has the value VRSEL specified.

### **Exclude from VRSEL**

Specifies that any tape data sets created on a tape volume set which is managed by the VRSEL retention method can be excluded from VRSEL inventory management. The data set VRSELEXCLUDE attribute is set for all data sets on volumes managed by the EXPDT retention method and is not affected by this support.

- Length: 1 character
- Type: Input
- Required: No

Valid values:

- Y - YES, if the data set is on a volume managed by the VRSEL retention method, the data set is excluded from VRSEL inventory management processing.
- N - NO, no exclusion, used only for data sets on a volume managed by the VRSEL retention method.
- blank - value is not specified.

Default: blank

**Note:** Exclude from VRSEL is not allowed to be specified when Retention Method has the value EXPDT specified.

### **Retain While Cataloged**

Specifies the default WHILECATALOG value for data sets on volumes managed by the EXPDT retention method.

- Length: 12 characters
- Type: Input

- Required: No

Valid values:

- OFF - specifies that new data sets residing on volumes managed by the EXPDT retention method will by default have the WHILECATALOG attribute set to OFF. As a result, retention of these data sets depends only on their expiration date and not on their catalog status.
- ON - specifies that new data sets residing on volumes managed by the EXPDT retention method will by default have the WHILECATALOG attribute set to ON. As a result, these data sets will not be expired as long as they are cataloged, nor will they be expired prior to their expiration date. The expiration date of the new data sets will be set to the greater of the EXPDT value or (creation date+ CATRETPD hours).
- UNTILEXPIRED - specifies that new data sets residing on volumes managed by the EXPDT retention method will by default have the WHILECATALOG attribute set to UNTILEXPIRED. As a result, these data sets will expire after they are uncataloged or if the expiration date is reached. When DFSMSrmm determines that a data set is no longer cataloged, the DFSMSrmm resets the expiration date of the data set to a date equal to the date the data set was uncataloged plus the number of days specified by the CATLGDAY option. New uncataloged data sets with the default of WHILECATALOG(UNTXILEXPIRED) will expire in the number of hours specified by the CATRETPD option unless they are cataloged before they expire.
- blank – value is not specified.

Default: blank

## 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 “[Defining management class expiration attributes](#)” on [page 70](#) 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 on [page 72](#) shows several combinations of retention attributes used for space management processing. Use the highlighted values for each instance.

<i>Table 3. Comparing Retention Period Attributes</i>					
Attribute	Case 1	Case 2	Case 3	Case 4	Case 5
Retention Limit	0	<b>50</b>	100	NOLIMIT	100
Expire after Days Non-usage	<b>50</b>	50	50	50	50
Expire after Date/Days	<b>100</b>	100	100	100	<b>100</b>
RETPD/EXPDT	60	60	<b>60</b>	<b>60</b>	

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, 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*.

Page 2 of the **Management Class Define** panel 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.

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

Applies to non-VSAM data sets, to all VSAM data set types that are allocated in the extended format, and to VSAM LDS as zFS. 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.

For VSAM LDS as zFS, when the **zfsadm shrink** command is issued, space release will be carried out by zFS.

#### YI

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

## 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.

For VSAM LDS as zFS, when the **zfsadm shrink** command is issued, space release will be carried out by zFS if the data set is guaranteed space.

## CI

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

## N

No. No release of unused space.

### Note:

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, EOVS, or FEOVS request for the data set.
  - Another DCB is open for the data set.

For more information on partial release, refer to *z/OS DFSMS Using Data Sets*.

### Migration Attributes

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 MLO. The default is 2. (Refer to **GDS Management Attributes** 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.

A Level 2 Days Non-usage attribute is used to describe direct migration to the cloud storage specified in the **Cloud Name** field. It supports two values, 0 and NOLIMIT. NOLIMIT is default. When NOLIMIT is specified, the data set will not be migrated to cloud storage and will be migrated based on the value of **Level 1 Days Non-usage**. When 0 is specified, the data set will be migrated directly to the cloud specified by the **Cloud Name** field as long as the **Primary Days Non-usage** value is met and the data set still resides on Level 0. This means **Level 2 Days Non-usage** takes priority over **Level 1 Days Non-usage**.

**Restriction:** DFSMSHsm does not support movement of migration copies from ML1 or to ML2 cloud storage.

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, specify 0 in the **Level 1 Days Non-usage** field and NOLIMIT in the **Level 2 Days Non-usage** field. Otherwise, the data sets must be first migrated to Level 1 (the days spent on Level 0 count towards eligibility of moving to Level 2).

New migration actions can also be made based on the size of a data set (in tracks) as long as the **Primary Days Non-usage** criteria is met. **Size Less Than** or **Equal To** and **Size Greater Than** are optional fields that specify a size threshold. If the current size of the data set meet the threshold for **Size Less Than** or **Equal To** or **Size Greater Than** fields, then the specified action is taken. For example, if the Primary Days Non-usage criteria is met and the **Size Greater Than** field is set to 200 tracks, then a data set of 250 tracks will be migrated to location specified by the **Action** field that is associated with the **Size Greater Than** field.

#### **Primary Days Non-usage**

Specifies the minimum number of days that are required to elapse for an unreferenced data set to be eligible for migration. Migration will take into effect when the necessary criteria are met and there is sufficient space on the volume. Default value is set to 2.

#### **Level 1 Days Non-usage**

Specifies the minimum number of nonused days for a data set before it becomes eligible for migration from Level 1 to Level 2. This includes nonused days in Level 0. Accepted values are 0 to 9999, or NOLIMIT. 0 indicates that the data set is migrated directly from Level 0 to ML2. If the data set was already migrated to Level 1, then it is immediate migrated to Level 2. A value of 1-9999 indicates that the data set may be migrated first from Level 0 to Level 1. A value of NOLIMIT indicates that the data set that migrates to Level 1 will not be migrated to Level 2, meaning there is no limit to how long the data set will stay in Level 1. Default value is set to 60.

#### **Level 2 Days Non-usage**

Specifies a direct migration to cloud storage. Current possible values are 0 and NOLIMIT. A blank value also indicates NOLIMIT. A value of NOLIMIT indicates a data set will be migrated to Level 2 based on the value of **Level 1 Days Non-usage**. A value of 0 for this attribute indicates the data set will be migrated to the cloud storage specified by the **Cloud Name** field as long as the **Primary Days Non-usage** value is met and the data set still resides on Level 0. The **Level 2 Days Non-usage** processing takes precedence over the **Level 1 Days Non-usage** value. Default value is NOLIMIT.

#### **Command or Auto Migrate**

Specifies whether a data set is eligible to be migrated by both command and automatic processing, by command alone, or not at all. Possible values are BOTH, COMMAND, or NONE. BOTH indicates that both automatic and command migration can occur. COMMAND indicates that the data set cannot migrate automatically, but a migrate command for individual data sets can occur, regardless of the Primary Days Non-usage value. NONE specifies that the data set cannot migrate. Default value is BOTH.

#### **Size Less Than or Equal To and Size Greater Than**

Specifies the migration target based on size and action fields. **Size Less Than** or **Equal To** and **Size Greater Than** can have possible values of 1 to 4294967295, or blank. Blank will indicate that the migration based on size feature will not be used. The values 1 to 4294967295 are measured in TRACKS, and indicate the threshold values for when a migration will occur. If a threshold value AND Primary Days Non-usage value criteria are met, then the data set will migrate based on the specified ACTION of that threshold. Both Size Less Than or Equal To and Size Greater Than have their own ACTION field. Default value for size thresholds are blank.

The Action fields for **Size Less Than** or **Equal To** and **Size Greater Than** specify the migration action that will occur when the data set size threshold is met. Default value is blank. Possible Action values are:

##### **NONE**

Data set should not be migrated. May be useful if you do not want to migrate data sets of below or above a certain size.

##### **ML1**

Data set should be migrated to the ML1 tier.

##### **ML2**

Data set should be migrated directly to the ML2 tier regardless of the values for **Level 1 Days Non-usage**.

**MIG**

Data set should be migrated to ML1, or ML2 according to the value of **Level 1 Days Non-usage**.

**TRANS**

Data set should have class transition processing performed on it.

**CLOUD**

Data set should be migrated to the cloud storage named in the **Cloud Name** field.

**Note:** The Size Less Than or Equal To and Size Greater Than attributes are only used during space management.

**Cloud Name**

Specifies the target for cloud storage migration. The SMS Cloud construct with the same name must also exist and the password for the cloud must be defined to DFSMSHsm. If **Level 2 Days Non-usage** or CLOUD as an Action to the data set thresholds, **Size Less Than** or **Equal To and Size Greater Than**, then this **Cloud Name** attribute is required. If a data set is to be migrated to the cloud, then the volume that the data set resides on must be connected to cloud storage. Default value is blank.

**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.

**Tip:** 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 whether 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, EXPIREDDATASETS, 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 are used to process the data set.

During automatic primary space management, GDSs that were 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.

## Defining management class backup attributes

Page 3 of the Management Class Define panel contains the management class backup attributes. (For objects, see [“Defining class transition attributes”](#) on page 79.)

### Backup Frequency

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

For DFSMSdftp OAM object support, if Auto Backup = 'Y', the *Backup Frequency* attribute specifies when the first backup copy of an object is to be written. If 0 is specified, then OAM will schedule a backup copy at the time the object is initially stored. If non-zero is specified then OAM will schedule backup copy(s) to be written during the OAM Management Component (OSMC) storage group processing cycle. The default is 1.

### Number of Backup Vers

Specifies 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.

Creating a new backup version when the number of backup versions already equals the value specified for the appropriate *Number of Backup Vers* attribute (Data Set Exists) causes the oldest version of the appropriate type to be 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 CBROAMxx 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 Ver

Indicates how many days to keep the most recent backup version of a deleted data set, starting from the day DFSMSShsm 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 Vers

Indicates how many days to keep backup versions other than the most recent one, starting from the day backups were created. It applies only when more than one backup version exists, and when a data set has low activity. This attribute applies whether the data set has been deleted or not. The default is 30.

The number of extra versions is the number of backup versions minus one. If you specify 1 for *Number of Backup Vers*, there are no extra versions. For example, if you specify 3 for *Number of Backup Vers (Data Set Deleted)*, the number of "extra" versions for deleted data sets is 2. These 2 versions are managed according to the *Retain Days Extra Backup Vers* attribute. Any other versions that may have existed when the data set was deleted will be deleted the next time EXPIREBV is processed.

### Admin or User Command Backup

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**

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**

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.

**AR|R**

indicates DFSMSdss keyword CONCURRENT(ANYREQ)

**VR**

indicates DFSMSdss keyword CONCURRENT(VIRTUALREQ)

**CR**

indicates DFSMSdss keyword CONCURRENT(CACHEREQ)

**AP|P**

indicates DFSMSdss keyword CONCURRENT(ANYPREF) or CONCURRENT

**VP**

indicates DFSMSdss keyword CONCURRENT(VIRTUALPREF)

**CP**

indicates DFSMSdss keyword CONCURRENT(CACHEPREF)

**S**

indicates not specifying CONCURRENT keyword to DFSMSdss. This is the default.

**Recommendation:** 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, on which you can specify the object attributes.



## Defining class transition attributes

Pages 4 and 5 of the Management Class Define panel contain the class transition attributes. The class transition attributes apply to both OAM and HSM class transition processing.

### **Time Since Creation Years, Months, or Days**

Indicate the time since the creation date that must pass before transition occurs.

### **Time Since Last Use Years, Months, or Days**

Indicate the time since the last reference date that must pass before transition occurs.

### **Periodic**

Indicates a time based on the calendar at which transition occurs.

**Restriction:** 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.

### **Monthly On Day**

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.

### **Quarterly On Day or In Month**

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.

### **Yearly On Day or In Month**

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.

### **Transition Copy Technique**

Specify which copy technique should be used for the class transition of data associated with this management class.

#### **FRP**

Fast replication preferred

#### **FRR**

Fast replication required

#### **STD**

Standard (the default)

#### **PMP**

Flashcopy Preserve Mirror preferred

#### **PMR**

Flashcopy Preserve Mirror required

#### **FCX**

Remote pair FlashCopy for XRC

For more information about fast replication, FlashCopy, Preserve Mirror and other copy services functions, refer to [\*z/OS DFSMS Advanced Copy Services\*](#).

### **Serialization Error Exit**

Specify the application or user exit to invoke when there is a serialization error. This is used by HSM during a class transition to notify DFSMSdss of which application needs to be invoked when a serialization error occurs.

## DB2

Db2 is invoked with the Db2 CAF interface. Db2 closes the data set. If there are no Db2 transactions in progress and the data set is successfully closed and unallocated, then the data set is exclusively serialized. If exclusive access is obtained, then the data set is transitioned. After the data set has moved, Db2 is reinvoked to allocate and open the data set. If the serialization cannot be obtained or there are active Db2 transactions, the transition fails. Only Db2 objects should be assigned to a management class with this setting.

## CICS

CICS is invoked with EXCI. CICS makes the data set unavailable for use by CICS and closes all files open to the data set. If these steps are successful, the data set is exclusively serialized and then transitioned. After the data set has moved, CICS is reinvoked to enable the CICS files to use the data set and make available the data set to be used by CICS. If the serialization cannot be obtained on the second attempt, then the transition fails. Only CICS data sets should be assigned to a management class with this setting.

**Note:** See [Appendix A, “Sample batch job for CICS definitions,” on page 393](#) for a sample batch job that will define the group, transaction, program, session and connection to allow the CICS server program to process the closing and opening of files.

## ZFS

z/OS File System causes an UNMOUNT to be issued. If the data set is successfully unmounted, then the data set is exclusively serialized. The unmount fails if the file system is currently accessing the data set. If exclusive access is obtained, then the data set is transitioned. After the data set has moved, zFS is reinvoked to mount the data set. Only zFS data sets should be assigned to a management class with this setting.

## EXIT

A user exit is invoked to unserialize the data set. The exit is invoked twice: initially to unserialize the data set, and a second time after the transition in order to reserialize the data set. The transition is performed if the data set can be exclusively serialized after the user exit has been initially invoked. A valid exit should be in place before this option is specified. No default exit is provided. The exit invoked is ADRDYEXT. See [z/OS DFSMS Installation Exits](#) for information about how this exit can be used.

## NONE

The transition should fail with no additional action. This is the default. As a default, DFSMSHsm does not issue error messages for data sets that fail serialization.

## Defining aggregate backup attributes

Page 6 of the Management Class Define panel contains the aggregate backup attributes. The SCDS Name field specifies the name of the SCDS into which the management class is defined.

### # Versions

Specifies the number of versions to be maintained.

### Retain Only Version

Indicates how long the most recent backup version of an aggregate group is kept.

### Unit

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.

### Retain Extra Versions

Indicates how long to keep backup versions of an aggregate group that precede the most recent version.

### Unit

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.

### Copy Serialization

Specifies whether aggregate backup should continue if an enqueue failure is encountered.

### Abbackup Copy Technique

Specifies whether concurrent copy is to be used to back up aggregates of data sets associated with this management class. The default value is STANDARD. Valid values for this field are:

#### CP

CACHE PREFERRED

#### CR

CACHE REQUIRED

#### P

CONCURRENT PREFERRED

#### R

CONCURRENT REQUIRED

#### S

STANDARD

#### VP

VIRTUAL PREFERRED

#### VR

VIRTUAL REQUIRED

## Assigning management classes

---

You can assign management classes through the management class ACS routine, by explicit specification or as part of a 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.

The management class determined by the management class ACS routine takes precedence over a management class that was specified explicitly. You can specify a management class explicitly with:

- 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 70](#) 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 13, “Defining ACS routines,”](#) on page 149.

## 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 components”](#) on page 208.

---

## Chapter 9. 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 that supports concurrent copy, SnapShot, or FlashCopy.
- VSAM record-level sharing between different systems with storage classes defined to use the coupling facility (CF).

This topic describes storage classes and shows you how to define them through ISMF.

**Restriction:** *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.

With STORCLASS=DUPT@SMS and DISP=NEW (file sequence > 1) and VOL=SER=nnnnnn specified in the JCL, you can also access a tape outside of an automated tape library for output if you have another tape with the same volser in a system-managed library. Specifying DUPT@SMS indicates that the DISP=NEW request is to be non-system managed. This directs the allocation of the duplicate volume to a stand-alone device. For DISP=NEW (file sequence 1), the assignment of a storage class when the automatic class selection (ACS) routines are run determines if the initial request is system-managed.

---

### 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.

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 [Chapter 21, “SETCACHE functions and device information,”](#) on page 309. 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 by using the storage group ACS routine.

## Storage classes for objects

For OAM, storage classes separate performance objectives from physical storage and determines where the object exists in the storage hierarchy.

OAM uses storage classes to logically represent the level of service required by an object. This might result in placement of the object on one of these types of physical storage in its object storage hierarchy:

- Direct access storage devices (Db2 object tables)
- File system (NFS or zFS)
- Optical volumes
- Tape volumes
- Cloud storage

The default storage class for an object is defined in the Db2 Collection Name Table entry for the object collection to which the object belongs. The default storage class is assigned by the storage class ACS routine when the first object is stored in the collection. If the Db2 Collection Name Table entry for the collection does not exist, it is created during a first-time store to a new collection by 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. For related topics, see:

- [“Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF”](#) on page 200 for information on generating lists of storage classes

- “Altering storage classes” on page 205 for information on altering a storage class
- [z/OS DFSMS Using the Interactive Storage Management Facility](#) for details on displaying storage class information

Perform the following steps to define storage class attributes.

1. Select option 5, Storage Class, from the ISMF Primary Option Menu for storage administrators. This displays the Storage Class Application Selection panel.

- 
2. Supply values on the Storage Class Application Selection panel:

**CDS Name**

This is the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active'. This represents the currently active configuration. You cannot define storage classes for the active configuration.

**Storage Class Name**

This is the name of the storage class. ISMF primes the Storage Class Name field with the name last used within ISMF.

**Option**

Select option 3, Define. This displays the Storage Class Define panel.

- 
3. Supply values on the Storage Class Define panel.

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.

**Description**

Is an optional 120-character description of the storage class.

For more information on the storage class attributes, see:

- [“Defining performance objectives” on page 85](#)
- [“Defining availability” on page 92](#)
- [“Defining accessibility” on page 93](#)
- [“Defining the guaranteed space attribute” on page 95](#)
- [“Defining guaranteed synchronous write” on page 98](#)
- [“Defining use of the coupling facility for VSAM record-level sharing” on page 98](#)
- [“Defining when to disconnect the sphere” on page 99](#)

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. The fields are:

- Direct Millisecond Response
- Direct Bias
- Sequential Millisecond Response
- Sequential Bias

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. For guidance on specifying MSR and bias values in the storage class, see [Processing Techniques in z/OS DFSMS Using Data Sets](#).

For the Direct Millisecond Response under the DFSMSHsm class, transition in the Storage Class will function as follows:

1. From 1 to 4,  
Class transition must occur on a SSD device.
2. Greater than 4,  
Class transition must occur on a non-SSD device.
3. Blank (default)  
Class transition can occur on any device.

## 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 [Table 4 on page 87](#) 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.

In table Table 4 on page 87, Read is abbreviated as R and Write is abbreviated as W.

Table 4. MSR Capabilities														
Device	Controller	Uncached <sup>1</sup>				Cached				Cached DASD Fast Write				IART <sup>2</sup>
		Direct		Sequential		Direct		Sequential		Direct		Sequential		
		R	W	R	W	R	W	R	W	R	W	R	W	
3350 3375	3380-1 <sup>4</sup>	30	30	30	30	30	30	30	30	30	30	30	30	0
3380	3380-3 <sup>4</sup>	25	25	25	25	25	25	25	25	25	25	25	25	0
	3380-13 <sup>3</sup>	25	25	25	25	14	25	14	25	14	25	14	25	0
	3380-23 <sup>3</sup>	25	25	25	25	10	25	10	25	10	25	10	25	0
3380 3390-1 3390-2 3390-3	3390-2 <sup>4</sup>	25	25	25	25	25	25	25	25	25	25	25	25	0
	3390-3/6	25	25	25	25	10	25	10	25	10	6	10	6	0
3390-9	3990-2 <sup>4</sup>	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 <sup>4</sup>	25	25	25	25	25	25	25	25	25	25	25	25	0
	9343-xCx <sup>3</sup>	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 <sup>4</sup>		600	620	230	250	600	620	230	250	600	620	230	250	10
3995-153 <sup>4</sup>		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
2107 solid state drives		1	6	10	6	1	6	10	6	1	6	10	6	0
<b>Note:</b>  1. Uncached values of a device are always used if an MSR of 999 is coded. 2. Initial access response time (IART) values are in seconds. All others are in milliseconds. 3. Device does not support DASD Fast Write. 4. Device does not support either CACHE or DASD Fast Write.														

From the numbers included in Table 4 on page 87, 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.

To direct allocation to a solid state drive, specify a direct MSR of 1 and a direct bias of R. Do not specify a value for sequential MSR or sequential bias. To direct storage allocation away from solid state drives, specify a direct MSR of 10 and a direct bias of R. Do not specify values for sequential MSR or sequential bias.

### ***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, as well as the performance capabilities of the device. For a list of the MSR and bias fields, refer to “Defining performance objectives” on page 85.

- Cache usage attribute.

The data set might be assigned one cache usage attribute for sequential accessing mode and a different attribute for direct accessing mode. The possible values that can be assigned for cache usage are:

#### **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 5 on page 89](#).

- DASD fast write usage attribute:

The data set might be assigned 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, because 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 5 on page 89 can help you determine which setting (May, Must, Never) is set by DCME.

Table 5. D/T3990 SMS Cache Candidate Tokens for Sequential and Direct Requests

Direct/Sequential MSR	Dir/Seq Bias	Direct Cache Token		Sequential Cache Token	
		Read	Write "3" on page 90	Read	
999	any	Never	Never	Never	Never
blank	blank	May	May "2" on page 90	May "1" on page 90	May
MSR < UC <sub>r</sub> "4" on page 90 and MSR < UC <sub>w</sub> "5" on page 90	blank	Must	May "2" on page 90	Must	May
MSR < UC <sub>r</sub> and MSR >= UC <sub>w</sub>	blank	Must	May "2" on page 90	Must	May
MSR >= UC <sub>r</sub> and MSR < UC <sub>w</sub>	blank	May	May "2" on page 90	May "1" on page 90	May
MSR >= UC <sub>r</sub> and MSR >= UC <sub>w</sub>	blank	May	May "2" on page 90	May "1" on page 90	May
blank	Read	May	May "2" on page 90	May "1" on page 90	Never

Table 5. D/T3990 SMS Cache Candidate Tokens for Sequential and Direct Requests (continued)

Direct/Sequential MSR	Dir/Seq Bias	Direct Cache Token		Sequential Cache Token	
		Read	Write <a href="#">“3” on page 90</a>	Read	
MSR < UC <sub>r</sub> and MSR < UC <sub>w</sub>	Read	Must	May <a href="#">“2” on page 90</a>	Must	Never
MSR < UC <sub>r</sub> and MSR >= UC <sub>w</sub>	Read	Must	May <a href="#">“2” on page 90</a>	Must	Never
MSR >= UC <sub>r</sub> and MSR < UC <sub>w</sub>	Read	May	May <a href="#">“2” on page 90</a>	May <a href="#">“1” on page 90</a>	Never
MSR >= UC <sub>r</sub> and MSR >= UC <sub>w</sub>	Read	May	May <a href="#">“2” on page 90</a>	May <a href="#">“1” on page 90</a>	Never
blank	Write	May	May <a href="#">“2” on page 90</a>	May <a href="#">“1” on page 90</a>	May
MSR < UC <sub>r</sub> <a href="#">“4” on page 90</a> and MSR < UC <sub>w</sub> <a href="#">“5” on page 90</a>	Write	Must	Must	Must	Must
MSR < UC <sub>r</sub> and MSR >= UC <sub>w</sub>	Write	Must	May <a href="#">“2” on page 90</a>	Must	May
MSR >= UC <sub>r</sub> and MSR < UC <sub>w</sub>	Write	May	Must	May <a href="#">“1” on page 90</a>	Must
MSR >= UC <sub>r</sub> and MSR >= UC <sub>w</sub>	Write	May	May <a href="#">“2” on page 90</a>	May <a href="#">“1” on page 90</a>	May

**Note:**

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. UC<sub>r</sub> is uncached values read performance.
5. UC<sub>w</sub> is uncached values write performance.

See [Table 4 on page 87](#) 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 (IARS) 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, the IARS, OAM Sublevel (OSL), and sustained data rate (SDR) are applicable.

OAM uses IARS as follows:

- If the IARS value is 0, OAM writes to a Db2 table, a file system, or the cloud level depending on the OSL value as described in [“Planning storage groups for OAM object collections” on page 31](#).
- If the IARS 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 IARS 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 IARS 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, when more than one volume has been given the same preference weight, so SMS does not always select the volume that has the most free space.

You can specify `USE_MOST_FREE_SPACE (YES)` to allow the SMS randomization of volume selection to select a volume with the most sufficient free space when there is more than one volume that has the same preference of weight on the secondary or tertiary list. The default is `NO` and this parameter does not have any effect on the primary list.

See [“SMS volume selection for data set allocation” on page 101](#) 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 that use 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 (PS) 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 assumed device transfer rate to derive the number of stripes that a striped data set can have. The SDR field specifies the target throughput rate. Valid values for this attribute are blank or from 0 to 999. The maximum number of stripes for PS data sets is 59. For VSAM data sets, the maximum number of stripes is 16.

- 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 106](#).

See [z/OS DFSMS Using Data Sets](#) for more information on using SMS to assign storage classes for striped data sets.

## Object access

When IARS > 0, 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.

OAM uses the OSL field to determine the level/sub level as follows:

- When IARS = 0
  - OSL = 1 means disk sublevel 1 (DB2)
  - OSL = 2 means disk sublevel 2 (file system)
  - OSL = 3 means cloud level
- When IARS > 0 and SDR >= 3
  - OSL = 1 means tape sublevel 1
  - OSL = 2 means tape sublevel 2

OSL has no meaning when IARS > 0 and SDR < 3. The default value for OSL is 1.

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 that supports the point-in-time copy, by 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 enables the 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 Enterprise Storage Server with the FlashCopy service.

The IBM Enterprise Storage Server 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\*](#) for further information on the Enterprise Storage Server 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 time. Versioning is done by using the SnapShot feature of the RAMAC Virtual Array or the Enterprise Storage Server 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 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 Enterprise Storage Server 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 by SMS volume selection.
- Data set allocation can cross different SFIs by default.
- A subparameter, Single SFI, indicates whether data set allocation across different storage facility images (SFIs) is allowed.



### CONTINUOUS PREFERRED (P)

Point-in-time copy volumes are preferred over non-point-in-time copy volumes.

Data set allocation can cross different SFIs by default.

### 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 6 on page 94 identifies the values that you specify based on your specific accessibility requirements

When Single SFI is Y, SMS fails when allocating or extending the data set to the same SFI is not possible.

Single SFI option is not saved permanently with the other data set attributes in either a catalog or VTOC entry. Therefore, when Single SFI is set to Y in a storage class definition, it affects extending existing data sets defined with the storage class on the release 3.1 and up systems where the support is included. Extending existing data sets defined with the storage class on the pre-3.1 systems are not affected.

When Single SFI is set to Y and a data set was previously allocated across multiple SFIs, the SFI option has no effect.

The ACCESSIBILITY = CONTINUOUS attribute asks SMS volume selection to select point-in-time copy volumes only. However, the data set allocation can cross different SFIs, which makes the usability of FlashCopy is not possible because on some boxes (old DS8K, possibly current OEM), the SFI is considered to be separate, and the box doesn't allow FC to devices outside the SFI. On newer boxes, there are two SFIs in the box, which can have FC between them but the performance is better in the same SFI than it is going across SFIs.

The ACCESSIBILITY = CONTINUOUS PREFERRED attribute asks SMS volume selection to prefer point-in-time copy volumes. If there are not enough point-in-time copy volumes, the data set allocation may continue with other available volumes. The data set allocation can cross different SFIs, and the extend doesn't fail.

The system might allocate or extend a multi-volume data set to multiple SFIs if the storage class does not prevent it. After that, an authorized person might modify the storage class to have the single SFI attribute or might use IDCAMS ALTER to switch the storage class for the data set to be a storage class with the single SFI attribute. In these two cases, the single SFI feature has no effect.

Table 6. 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 nonaccessibility 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



Table 6. Combinations for Requesting Point-in-Time Copy Devices (continued)

If these are your choices for accessibility devices:			Specify the following values:
First Choice	Second Choice	Third Choice	
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.

With DISP=MOD, you can specify different primary space amounts to be allocated to each volume. To accomplish this you can specify multiple DD statements using the same data set name, each with different values for primary space. Refer to [“Preallocating space for multivolume data sets” on page 95](#) for examples.

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 Commands](#).

## Preallocating space for multivolume data sets

This topic shows scenarios for preallocating space for multivolume data sets.

### Allocating a multivolume data set on SMS-selected volumes

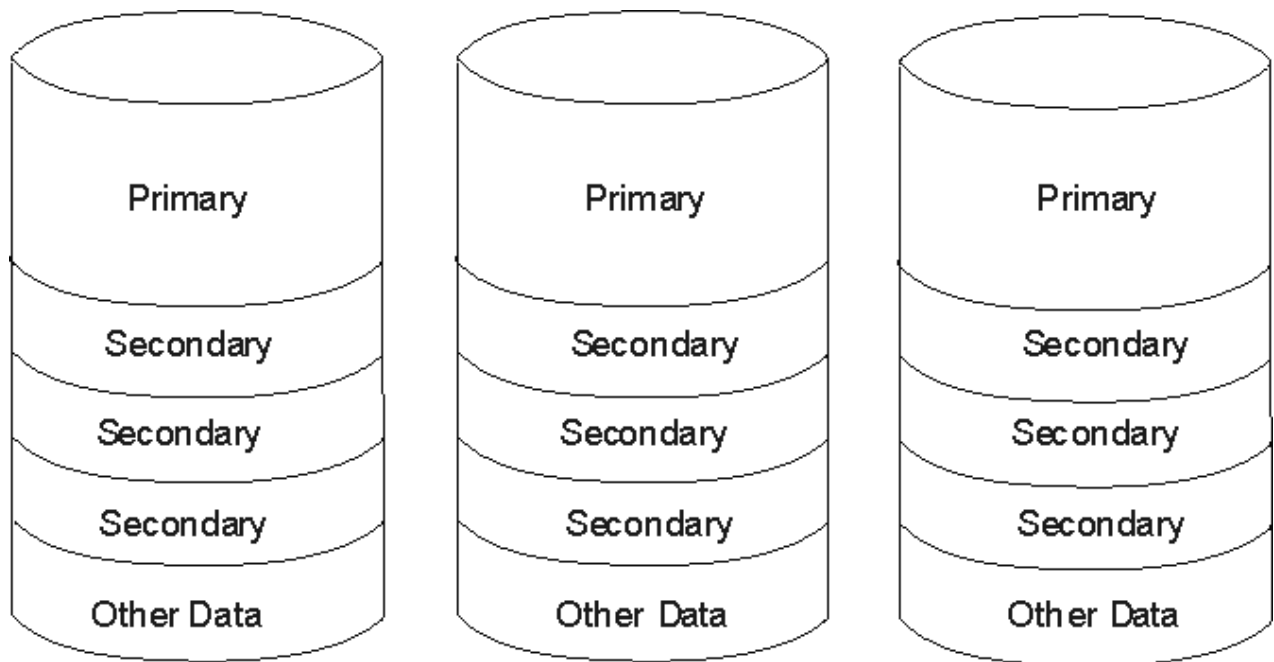
The following JCL allocates a multivolume data set on SMS-selected volumes. The storage class DBLOG must have Guaranteed Space=Y. This example allocates 100 MB on each of five volumes. When all of the

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.MULTIFILE,DISP=(,KEEP),STORCLAS=DBLOG,  
//      SPACE=(1,(100,25)),AVGREC=M,  
//      UNIT=(3380,5)
```

**Note:**

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 7 on page 96](#).



DA6S2002

*Figure 7. Allocation of Primary and Secondary Space for Multivolume Data Sets*

### **Allocating a multivolume data set with different primary allocation on each volume**

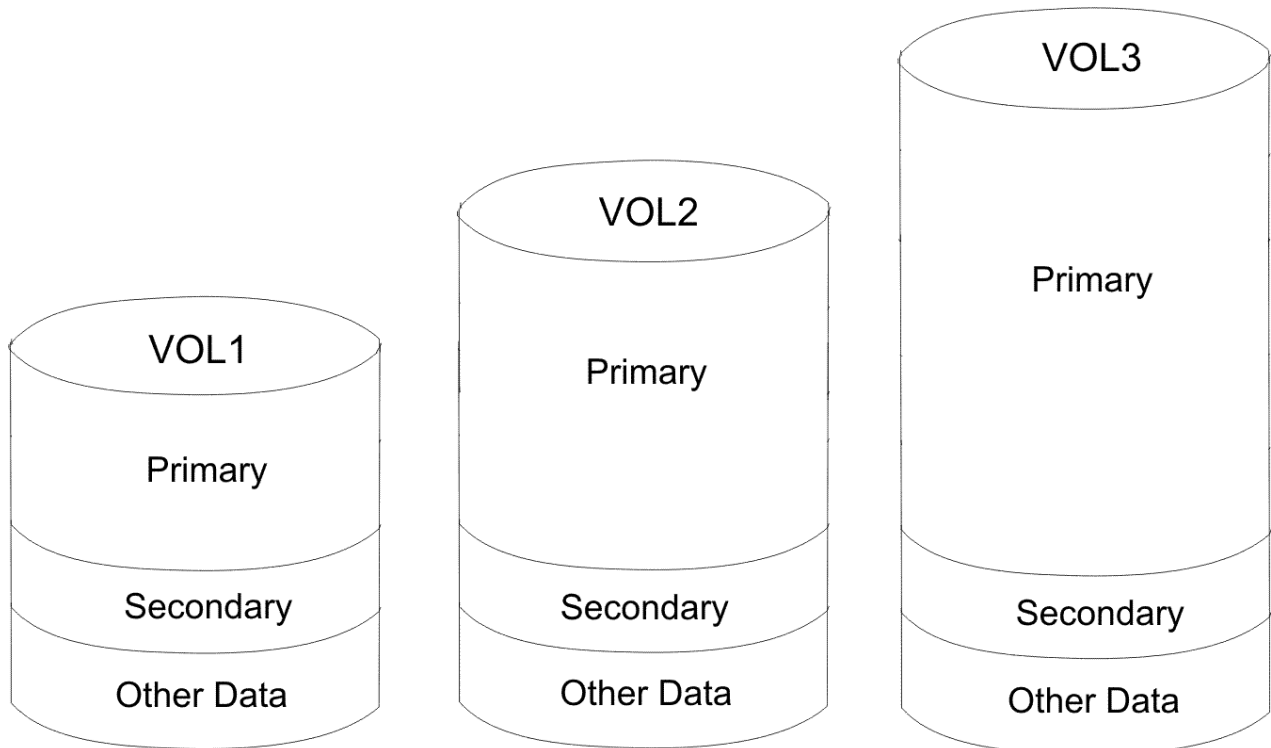
The following JCL allocates a multivolume data set on SMS-selected volumes with different primary allocation on each volume. The storage class DBLOG must have Guaranteed Space=Y. This example preallocates 10 cylinders on the first volume (VOL1), 20 cylinders on the second volume (VOL2) and 30 cylinders on the third volume (VOL3). When all of the primary space is used on a volume, the secondary space is allocated as required in extents on that volume. For example:

```
//DD1 DD DSN=DATASET1,DISP=(MOD,CATLG),SPACE=(CYL,(10,5)),VOL=SER=VOL1  
//DD2 DD DSN=DATASET1,DISP=(MOD,CATLG),SPACE=(CYL,(20,5)),VOL=SER=VOL2  
//DD3 DD DSN=DATASET1,DISP=(MOD,CATLG),SPACE=(CYL,(30,5)),VOL=SER=VOL3
```

**Note:**

1. After the 10 preallocated cylinders are used on VOL1, 5 cylinder extents of secondary space are allocated on this volume until the extent limit is reached or the volume is full.

2. If more space is needed, the 20 cylinders of preallocated primary space is used on VOL2. Secondary space is allocated on this volume, as required, until the extent limit is reached or the volume is full.
  3. If more space is needed, the 30 cylinders of preallocated primary space is used on VOL3. Secondary space is allocated on this volume, as required, until the extent limit is reached or the volume is full.
- 



*Figure 8. Allocation of Primary and Secondary Space for Multivolume Data Sets with Different Primary Allocations on Each Volume*

---

## 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

Specifying Guaranteed Synchronous Write can ensure that data has been physically written to disk in the event of an outage, however it may degrade performance due to less efficient buffering.

When a program issues a CHECK macro successfully when writing, it normally ensures that the data is safely on the medium. Writing to a PDSE or compressed format data set is an exception, and the access method buffers the data after the CHECK. This greatly improves performance but some applications require individual records to be physically on the disk at the time the program writes them, and not buffered by the access method. Issuing a successful STOW (for PDSE) or CLOSE macro ensures that all data is safely on the medium. A partially created member cannot be recovered because it has no name until a STOW macro is issued. If your program is writing a compressed format data set or opened a PDSE with the UPDAT option, then the guaranteed synchronized write option causes each CHECK macro to force synchronization of buffered data to the disk.

Specify a Y for synchronized write or an N for no synchronization.

Specifying Guaranteed Synchronous Write can ensure that data has been physically written to disk in the event of an outage, however it will degrade performance due to less efficient buffering. It is recommended that this option be set to NO unless you determine that a particular application requires that its data be hardened to disk every time a write is issued.

For more information see "Using the SYNCDEV Macro to Synchronize Data" in [z/OS DFSMS Using Data Sets](#).

## Defining use of the coupling facility for VSAM record-level sharing

You can associate a storage class with a CF cache set that is defined in the base configuration, thereby making any data set associated with the storage class eligible for VSAM record-level sharing. You can also assign a weight value to the data, so as to indicate its relative importance. You should be familiar with the information in the following topics:

- [Chapter 18, “Administering VSAM record-level sharing,” on page 235](#)
- [“SMS volume selection for data set allocation” on page 101](#)
- [“Using the multi-tiered storage group function” on page 109](#)

Specify values for the following attributes:

### Multi-Tiered SG

Specify Y if you want SMS to prefer the storage group sequence order as specified in the ACS storage group selection routines. ISMF primes the field with the default N, No.

### Parallel Access Volume Capability

Specify R for required, P for preferred, S for standard, and N for no preference. The default is N.

### CF Cache Set Name

Is the name of a CF cache set that is 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.

### CF Direct Weight or the CF Sequential Weight

Specify a weight attribute indicating the data's relative importance. 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 supports only the default value. All data is assigned a weight value of 6 regardless of the value you specify.

## Defining when to disconnect the sphere

You can use the Disconnect Sphere at CLOSE attribute to indicate whether the sphere should be disconnected upon closing the data set or stay connected for a short period of time.

Type Y to disconnect the sphere when the data set is closed. This may be beneficial in an environment with constrained buffer pool space when the same data sets are not repeatedly closed and re-opened. Type N to keep the sphere connected for a short time after the data set is closed. N is the default.

## Defining use of zHyperLinks

You can specify whether data sets associated with this storage class are eligible to use zHyperLinks for reading and writing. zHyperLinks dramatically reduce latency by interconnecting the z14 Central processor complex directly to the I/O bays in the IBM DS8880. This can improve application response time, without significant application changes.

**Note:** For the zHyperLink protocol to be enabled for I/O requests by the Db2 system, in addition to specifying a value of ENABLE, DATABASE, or ACTIVELOG for the ZHYPERLINK parameter, the SMS storage class of the Db2 data sets must be enabled for zHyperLink eligibility with the Integrated Storage Management Facility (ISMF).

zHyperLink technology is limited to 4 K CI for write, and the size specified in the current READLIMIT setting for read. When you specify ZHYPERLINK ENABLE or DATABASE option, zHyperLink read is enabled for Db2 objects with CI size less than or equal to the current READLIMIT setting. zHyperLink read to greater than the current READLIMIT setting is disabled by z/OS. For more information about READLIMIT, see [ZHYPERLINK](#) in *z/OS MVS Initialization and Tuning Reference*.

Not all applications may see a benefit from using zHyperLinks. Review the list of good versus bad candidates before deciding to enable zHyperLinks for a storage class.

Specify values for the following attributes:

### zHyperLink Read Eligibility

Specify Y to make this storage class eligible for zHyperLink reads.

Good candidates:

- Synchronous (blocking) reads where the data is likely to be in the DASD cache.

Bad candidates:

- Read requests where data is not likely to be in DASD cache
- Read-ahead or pre-staging of data.

### **zHyperLink Write Eligibility**

Specify Y to make this storage class eligible for zHyperLink writes.

Good candidates:

- Synchronous (blocking) writes that follow a log write pattern. Writes may wrap around to the beginning of the data set.
- Block sizes: 4 K or 4K+32-byte suffix for extended format.
- Data accessed via Media Manager interface.

Bad candidates:

- Random writes
- Format writes
- Lazy/delayed writes
- Large sequential writes
- Data sets that contain physical keys
- Data accessed with EXCP or EXCPVR.

## **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 13, “Defining ACS routines,”](#) on page 149.

## 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 components”](#) on page 208.

## SMS volume selection for data set allocation

When a data set is created, SMS follows a sequence of steps to place it on a volume. This section describes how SMS selects a volume based on the requirements of a data set. It explains conventional volume selection, striping volume selection and multi-tiered volume selection. Finally, it discusses why volume selection might fail.

**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.

SMS attempts to spread initial allocations evenly across similar volumes and storage groups provided by the ACS storage group selection routine unless you specify Multi-Tiered SG (Y) in the storage class. In that case, SMS performs an allocation using storage groups in the order in which they are specified in the ACS storage group selection routine. For more information, see [“Using the multi-tiered storage group function”](#) on page 109.

## Volume selection preference attributes

Table 7 on page 101 shows some of the preference attributes used in volume selection for allocation of a data set.

Table 7. Volume Selection Preference Attributes

Criteria	Preferences
<b>PCU SEPARATION</b>	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.
<b>EXTENT POOL SEPARATION</b>	Volume does not reside in the same extent pool that has allocated a data set from which this data set should be separated, as specified in the data set separation profile.
<b>VOLUME SEPARATION</b>	Volume does not contain a data set from which this data set should be separated, as specified in the data set separation profile.
<b>VOLUME COUNT</b>	Volume resides in a storage group that has enough eligible volumes to satisfy the requested VOLUME COUNT.
<b>PRIMARY THRESHOLD</b>	Volume has sufficient space in the target addressing space for the allocation amount without exceeding the storage group HIGH THRESHOLD value.
<b>SECONDARY THRESHOLD</b>	Volume has sufficient space for the allocation amount without exceeding the storage group HIGH THRESHOLD value.
<b>SMS STATUS</b>	Volume and its associated storage group SMS status are ENABLED.
<b>MULTI-TIERED STORAGE GROUP</b>	Volume resides in a storage group that will be selected in the order of specification.
<b>END-OF-VOLUME EXTEND</b>	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.

Table 7. Volume Selection Preference Attributes (continued)

Criteria	Preferences
<b>NONOVERFLOW</b>	Volume resides in a nonoverflow storage group.
<b>IART</b>	Volume is mountable, and an IART value greater than zero was specified in the storage class.
<b>FAST REPLICATION</b>	Volume is eligible for a fast replication request.
<b>CLUSTER</b>	Volume resides in the same cluster.
<b>EXTENT POOL</b>	Volume resides in a different extent pool (for striping).
<b>CONTROLLER</b>	Volume resides on a device connected to a different logical control unit (LCU) (for striping).
<b>EXTENDED ATTRIBUTE</b>	Volume has extended attributes (format 8 and 9 DSCBs) or optionally resides in extended addressing space.
<b>ACCESSIBILITY</b>	Volume resides in a control unit that supports ACCESSIBILITY, and the storage class ACCESSIBILITY value is PREFERRED.
	Volume resides in a control unit that does not support ACCESSIBILITY, and the storage class ACCESSIBILITY value is STANDARD.
<b>PARALLEL ACCESS VOLUME</b>	Volume supports the PAV specification in the storage class
<b>AVAILABILITY</b>	Volume resides in a control unit that supports AVAILABILITY and the storage class AVAILABILITY value is PREFERRED.
	Volume resides in a control unit that does not support AVAILABILITY and the storage class AVAILABILITY value is STANDARD.
<b>EXTENDED FORMAT</b>	Volume resides in a control unit that supports EXTENDED FORMAT and the data class IF EXT value is PREFERRED.
<b>MILLESECOND RESPONSE (MSR)</b>	Volume provides the requested response time that is specified or defaulted in the storage class DIRECT MSR or SEQUENTIAL MSR.
	Volume provides a faster response time than what is specified or defaulted in the storage class DIRECT MSR or SEQUENTIAL MSR.

The volume selection preference attributes in [Table 7 on page 101](#) are assigned a weight in the order shown (highest to lowest). SMS selects the volume with the highest cumulative preference weight.

## Conventional volume selection

Conventional volume selection is used for all nonstriped data sets. It is also used for data sets that were allocated with an SDR of zero or blank.

Conventional volume selection uses the criteria that are specified in the storage class and the data class as preference attributes for volume selection. SMS uses the preference attributes in the following situations:

- To classify the volume as *primary*, *secondary*, or *tertiary*. Only volumes that meet all of the requested attributes can be classified as primary volumes.
- To assign a selection weight to the volume.

SMS begins by classifying each volume in a storage group as primary, secondary, tertiary, or rejected.

### Primary

All the volumes in all the specified storage groups are candidates for the first, or primary, list.



**Exception:** When Multi-Tiered SG (Y) is specified in the storage class, only the volumes in the first storage group that is assigned by the ACS routine are eligible for the primary list. Also, volumes in Overflow and Extend storage groups are not eligible for the primary list.

The primary list consists of online volumes that meet all the requested preference attributes, are below their 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.

SMS chooses from the candidates on this list. It prefers 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). If no devices in the storage groups meet all of the primary list criteria, no volumes will appear on the primary list. It is possible that, even though there are devices on this primary list, the data set cannot be successfully allocated to any of these devices. That is, 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 selection preference attributes listed in [Table 7 on page 101](#), 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. All fixed DASD volumes will be placed on the secondary list, which uses randomization between similar volumes.

## 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.

If a data set cannot be allocated on a primary volume, SMS continues to evaluate volumes on the secondary list until allocation is successful. SMS evaluates volumes in the secondary list based on volume characteristics, performance, available space, and system accessibility. For more information about volume selection preferences, see [Table 7 on page 101](#).

### Note:

- If no volumes meet the space criteria, they do not qualify for the primary list. This means that they will be placed on the secondary or tertiary list.
- With IART of nonzero, no eligible volumes are placed in the primary list. All eligible volumes are placed in the secondary or tertiary list.
- When in the secondary or tertiary list, SMS assigns the different preference of weight to a volume.
  - If all the volumes on the secondary or tertiary list are still under the high allocation threshold after the requested space is allocated, SMS assigns the higher preference of weight for unallocated volumes to a job or subsystem.
  - Otherwise, SMS assigns the higher preference of weight for the most free space and unallocated volumes to a job or subsystem.
  - After the weight is assigned, SMS uses the randomizing selection technique where SMS will randomize which volume to select for multiple volumes with the same weight. This randomizing selection technique may result in the selection of a volume with less space. Most often, the volume with the most free space will be preferred.

- You can specify `USE_MOST_FREE_SPACE(YES)`, when in the secondary or tertiary list, to allow SMS to select the volume with the most free space when there is more than one volume that has the same weight. SMS continues to use the randomizing selection technique among the volumes with the same weight and same most net free space.

### **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.

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.

### **Rejected**

Volumes that do not meet the required allocation specifications (`ACCESSIBILITY = CONTINUOUS`, `AVAILABILITY = STANDARD`, or `CONTINUOUS`, `ENABLED` or `QUIESCED`, `ONLINE...`) are marked rejected. Those volumes 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 that were requested for the data set. If you specify an extend storage group, the data may be extended to the specified extend storage group. For more information about how to specify an extend storage group by using the Extend SG Name parameter, see [“Defining a pool storage group” on page 35](#).

When the allocation fails due to insufficient space, users can, through use of Space Constraint Relief option in Data Class, indicate to SMS that they want to attempt a retry. SMS will retry the allocation request with Space Constraint Relief. For more information, see [“Space constraint relief” on page 109](#).

## **Types of allocation**

To improve performance, SMS limits the number of eligible volumes to those most likely to satisfy the request. Two types of allocations are used: *best-fit* and *non-best-fit*.

- The term *best-fit* implies that the allocation request was likely the result of a recall, copy, or restore operation. SMS might have used multiple volumes to satisfy the allocation request.
  - If the caller indicates that space constraint can be used, SMS will retry the best-fit allocation with the 5-extent limit removed, regardless of what the setting is in the data class `SPACE CONSTRAINT RELIEF` parameter.
  - The primary threshold preference attribute is always ON for all the eligible volumes regardless of what the setting is in the storage group Migration and Allocation Thresholds parameter. This ensures all the eligible volumes can participate in the volume selection process.

For best-fit allocations where the primary space can be spread out on multiple volumes, DADSM is called. The allocation continues until one of the following situations occurs:

- The primary space request is completed.
- The maximum number of volumes is reached but the requested space has not yet been completely received. SMS will reject the storage group after DADSM rejects 59 volumes for insufficient space.

The remaining volumes in the storage group are not considered for further selection because they are less preferred and are less likely to fulfill the allocation request.

- For non-best-fit allocation where the primary space must be gotten on one volume, SMS will exclude a volume from selection if the requested primary space exceeds its total capacity. For non-best-fit allocations that use *fast* volume selection, SMS performs volume selection from the prioritized list until 100 volumes have been rejected by DADSM for insufficient space. When that occurs, SMS will exclude, based on the volume statistics in the SMS configuration, all volumes that do not have enough free space. This *fast* volume selection approach can greatly reduce the number of candidate volumes,

and thus the number of retries. You can activate fast volume selection by using the FAST\_VOLSEL(ON) parameter in the IGDSMSxx parmlib member or the SETSMS FAST\_VOLSEL(ON) command.

## 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 7 on page 101](#), or for one of the following reasons:

- The volume is rejected. See [“Possible reasons for volume selection failure” on page 113](#) 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.

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 7 on page 101](#).

## 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-striped when migrated or backed-up, it follows the conventional volume selection path.

When striping volume selection is entered, the allocation request will not be eligible for Space Constraint Relief. The exception would be when the request is conditional (preferred) extended format and no available volumes meet the striping criteria, the allocation is attempted as non-striped, which means the conventional volume selection will be used.

Striping volume selection is very similar to conventional volume selection. Volumes that are eligible for selection are classified as primary and secondary, and assigned a volume preference weight, based on preference attributes. See [Table 7 on page 101](#) for more information on preference attributes.

Volumes are classified as follows:

### Primary

For each controller, SMS randomly assigns a single volume that meets all of the requested preference attributes as the primary volume.

## Secondary

For each controller, SMS classifies all eligible volumes other than the primary volume as secondary volumes.

SMS calculates the average preference weight of each storage group using the preference weights of the volumes that will be selected if the storage group is selected for allocation. Then, SMS selects the storage group that contains at least as many primary volumes as the stripe count and has the highest average weight. If there are no storage groups that meet these criteria, the storage group with the largest number of primary volumes is selected. If multiple storage groups have the largest number of primary volumes, the one with the highest average weight is selected. If there are still multiple storage groups that meet the selection criteria, SMS selects one at random.

After selecting a storage group, SMS selects volumes by their preference weight. Primary volumes are preferred over secondary volumes as they have a higher preference weight. Secondary volumes are selected when there are an insufficient number of primary volumes. If there are multiple volumes with the same preference weight, SMS selects one of the volumes at random.

Volumes that meet the requested MSR are preferred over volumes that do not meet the requested performance. A volume is considered to meet 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.

When allocating striped data sets, SMS prefers allocating stripes across extent pools. If this is not feasible, SMS allocates stripes across logical controllers (LCUs). When extending a stripe of a striped data set to a new volume, SMS prefers volumes in the extent pools that are not being used by other active stripes.

## Rules for striping volume selection

For both guaranteed and non-guaranteed space allocations, the following key rules apply for striping volume selection:

- Storage groups containing mixed device types are not considered.
- In either required or preferred extended format allocation requests, if the target number of stripes is not available, allocation will be attempted with a smaller number of stripes until either the allocation succeeds or all available volumes have been tried.
- Target volumes
  - If the SDR field is blank or 0, the target number of stripes is 1 for both guaranteed and non-guaranteed space requests and conventional volume selection is used. If the value in the SDR field is 1 resulting in a stripe count of 1, striping volume selection is used instead of conventional volume selection.
  - For non-guaranteed 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. 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 non-striped.
- The volume must be able to satisfy the primary space requested by the number of stripes.
- The maximum number of stripes (volumes) for VSAM data sets is 16. The maximum number of stripes for physical sequential (PS) data sets is 59.
- 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 component is 7257
- The minimum allocation is one track per stripe.
- The maximum allocation can exceed the 64K track limit.
- 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 EOVS perform this calculation.
- Non-VSAM multi-striped 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 non-guaranteed 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.
  - 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.

## 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.

## Cluster and storage facility image considerations

The DFSMSdss data set fast replication function requires that all volumes of a multi-volume data set reside in the same storage facility image (SFI). Data set fast replication is most efficient when both source and target data sets reside in the same cluster. To help you take advantage of data set fast replication, SMS volume selection:

- When allocating or extending an SMS-managed multi-volume data set that has point-in-time copy volumes requested, prefers candidate volumes that are in the same cluster, and, if that cannot be honored, prefers candidate volumes that are in the same SFI. Point-in-time copy volumes are requested when the ACCESSIBILITY parameter in the storage class is set to CONTINUOUS or CONTINUOUS PREFERRED.
- When allocating the target data set for data set fast replication, prefers volumes in the same cluster as the source data set, and, if that cannot be honored, prefers volumes in the same SFI as the source data set.

To make efficient use of the fast replication function, ensure that enough volumes are in one SFI to accommodate the likely increase in allocations to volumes in these storage groups and SFIs. For optimum efficiency, ensure that enough volumes are in one cluster to accommodate the likely increase in allocations to volumes in these storage groups and clusters.

The preference for volumes that are in the same cluster or SFI applies to all multi-volume allocations with ACCESSIBILITY = CONTINUOUS or CONTINUOUS PREFERRED. However, there are a few situations where SMS volume selection does not take the cluster or SFI attribute into consideration. These situations are:

- The volumes that are specified by the user for a guaranteed space request are not in the same SFI.
- No clusters or SFIs have a sufficient number of volumes to meet the number of volumes required for allocation.
- No clusters or SFIs have a sufficient number of unique controllers to meet the stripe count for a striping allocation.
- During extend processing, the volumes that the data set resides on are not in the same SFI.
- Space Constraint Relief (SCR) processing is in effect.

SMS volume selection does not take the cluster attribute into consideration if one of the following conditions is true:

- The hardware server independent attribute is set.
- The DFSMSdss 'nocopy' operation is invoked.

- The SYS1.PARMLIB(IGDSMSxx) parameter, BYPASS\_CLUSTER\_PREFERENCING, is set to YES.

## 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

## 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. For more information about space constraint relief, see [“Specifying attributes to handle space constraints during allocation” on page 126](#).

## Using the multi-tiered storage group function

**Before you begin:** Be familiar with conventional volume selection for allocation, which is described in [“SMS volume selection for data set allocation” on page 101](#).

You can specify Multi-Tiered SG (Y) in the storage class to enable the multi-tiered storage group function. When you specify this function, SMS allocates to enabled pool storage groups in the order they are specified in the ACS storage group selection routines.

**Example:** Multi-Tiered SG (Y) is specified in the storage class. An ACS storage group routine assigns the following three similar storage groups:

```
SET &STORGRP = 'SG1', 'SG2', 'SG3'
```

**Result:** SMS selects the enabled volumes that are below the high threshold in SG1 before selecting volumes in the SG2 storage groups that are listed second, third, and so forth, can also be used depending on the status of the underlying volumes when:

- All SG1 enabled volumes exceed high threshold, then SMS selects the enabled volumes in SG2.
- All SG1 and SG2 enabled volumes exceed the high threshold, SMS selects the enabled volumes in SG3.
- All enabled volumes exceed the high threshold, SMS selects the quiesced volumes in the same storage group order.

Only volumes that reside in the first storage group assigned by the ACS routines (SG1 in the example) are eligible for the primary list.

## Using the parallel access volume option

**Before you begin:** Be familiar with conventional volume selection for allocation, which is described in [“SMS volume selection for data set allocation” on page 101](#).

If the Enterprise Storage Server (ESS) PAV option is enabled, you can use the DFSMS storage class parallel access volume (PAV) option to ensure that data sets that require high performance are only allocated to volumes on which the ESS PAV option is enabled.

The DFSMS PAV capability option includes the following settings with these results:

**Required**

Only volumes with the PAV feature enabled are selected.

**Preferred**

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

**Standard**

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

**Nopreference**

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

## Planning to use the PAV option in the storage class

**Before you begin:** All storage classes defined by your installation are automatically assigned the default PAV value of Nopreference, which means that all PAV and non-PAV volumes are treated equally. To force volume selection to differentiate between them, you need to modify the PAV option in the storage class.

The following list comprises the planning steps that you must complete before you can successfully implement the PAV option in the storage class:

1. Verify that ESS devices have the PAV capability enabled using the DEVSERV command.

Example: `DS QPAVS,devnum`

For information about using the DEVSERV command, see [z/OS MVS System Commands](#).

- 
2. Determine which data sets require high performance. You could assign a PAV value of Required or Preferred to those data sets.

Data sets that require high performance could include frequently accessed data sets and critical data sets. For example, if certain jobs are taking too long because data sets used by these jobs are on non-PAV volumes, you might allocate those data sets on PAV volumes to determine if that improves their performance.

- 
3. Decide whether to assign the PAV option to existing storage classes. If you do not specify the PAV option in the storage class, the default value is Nopreference.

- 
4. Decide whether to create any new storage classes that use the PAV option.

Now you are ready to implement the SMS PAV option in the storage class.

## Implementing the PAV option enhancement

**Before you begin:** Ensure that your system is running at z/OS V1R6.

**Migration actions:** If you do not plan to exploit the PAV volume selection capability, you do not need to perform any migration actions. However, if you plan to exploit the PAV capability, you must make changes to the SMS configuration.

Perform the following steps to implement the PAV volume selection capability in the storage class:



1. Modify the PAV option in the SMS storage classes that are assigned to data sets that require high performance.

For the detailed steps, see [“Modifying storage classes to enable the PAV option” on page 111.](#)

---

2. Define new storage classes with a value other than NOPREFERENCE specified for the PAV option.

For the detailed steps, see [“Defining new storage classes with the PAV option enabled” on page 111.](#)

**Result:** Now you have more control over the volume selection process for a SMS-managed data set, depending on whether a volume is PAV enabled. You can update automatic class selection (ACS) routines to use the new or modified storage classes with the PAV capacity enabled.

**Restriction:** Striped data sets and best-fit data set allocations support the PAV option in a limited way. If PAV capability is required, all volumes that do not have this capability are rejected. However, if you specify a PAV capability of Preferred, Standard, or Nopreference for a striped or best-fit data set, this PAV option is ignored.

## Modifying storage classes to enable the PAV option

Perform the following steps to set up the PAV option in existing storage classes:

1. Use ISMF to access the Storage Class application.
- 

2. Select option 4, Alter a Storage Class.
- 

3. On page 2 of the Storage Class Alter panel, set the Parallel Access Volume Capability value to one of the following values:

- R (required)
  - P (preferred)
  - S (standard)
  - N (no preference)
- 

4. Activate the updated configuration using one of the following methods:

- a. On the CDS Application Selection panel in ISMF, complete one of the following actions:

- Select the Activate option, or
- Enter the ACTIVATE operator command on the command line.

Example: `ACTIVATE SCDS dsname`

- b. From the operator console, enter the SETSMS command.

Example: `SETSMS SCDS dsname`

**Result:** The PAV option is enabled for this storage class.

**Recommendation:** When you modify an existing storage class and activate the configuration, you are implicitly modifying the volume selection process for all data sets to which this storage class is assigned. If you do not want the "modified" storage class to apply to all data sets that it is being assigned to, you will need to change the storage class ACS routines.

## Defining new storage classes with the PAV option enabled

Perform the following steps to define new storage classes with the PAV option enabled:

1. Use ISMF to access the Storage Class application.
-

2. Select option 3, Define a Storage Class.

---

3. On page 2 of the Storage Class Define panel, set the Parallel Access Volume Capability value to one of the following values:

- R (required)
  - P (preferred)
  - S (standard)
  - N (no preference)
- 

4. Complete the remaining entries on the Storage Class Define option panel.

---

5. Update the ACS routines to include the new storage class.

For more information, see [Chapter 13, “Defining ACS routines,” on page 149 in the \*z/OS DFSMSdfp Storage Administration\*](#).

---

6. Activate the updated configuration using one of the following methods:

a. On the CDS Application Selection panel in ISMF, complete one of the following actions:

- Select the Activate option, or
- Enter the ACTIVATE operator command on the command line.

Example: `ACTIVATE SCDS dsname`

b. From the operator console, enter the SETSMS command.

Example: `SETSMS SCDS dsname`

**Result:** The new storage class is activated.

**NaviQuest tip:** You can specify the PAV option using the NaviQuest sample JCL, ACBJBAS1, to define, alter, or display a storage class.

## Displaying information about the PAV capability of a volume

Perform the following steps to display information about the PAV capability of a volume:

1. Use ISMF to access the Storage Class application.

---

2. Select option 2, Display a Storage Class. Go to page 2 of the Storage Class Display panel.

The Parallel Access Volume Capability field displays one of the following values:

- REQUIRED
- PREFERRED
- STANDARD
- NOPREFERENCE

You also can obtain information about the PAV capability of a volume using the following ISMF panels:

- Storage Class List (Parallel Access Vol field)
- Storage Class List Print (Parallel Access Vol field)
- Storage Class Sort Entry (Parallel Access Vol field)
- Storage Class List View (Parallel Access Vol field)

## Diagnosing problems with the PAV option

The following SMS messages indicate that volumes are rejected or not used because they do not allow the PAV requirement:

- IGD17268I
- IGD17269I
- IGD172791

ISMF issues the following new message if an invalid value is entered for the PAV option on the Storage Class Define/Alter panel:

- DGTSC084

For more information about these error messages, refer to [\*z/OS MVS System Messages, Vol 4 \(CBD-DMO\)\*](#) and [\*z/OS MVS System Messages, Vol 8 \(IEF-IGD\)\*](#).

## 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 resides in a copy pool backup storage group.
- 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 format 1 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 affect 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.

## Defining secondary lock tables

---

You can define multiple secondary VSAM RLS lock structures to be associated with a single SMS storage class. A secondary lock structure is identified by an SMS storage class attribute called a lock set. You can define up to 256 lock sets per sysplex. Each lock set can contain a single lock structure name. When an application opens a VSAM data set, RLS processing checks the storage class defined for the data set to determine which lock structure to use. If the storage class specifies a secondary lock structure, RLS processing uses the secondary lock structure to serialize access to records in the data set.

If you do not define secondary lock structures, VSAM RLS uses the primary DFSMS CF lock structure, the IGWLOCK00 lock table, to hold the record locks. However, using a single lock structure forces DFSMS to combine all locking information for all instances of the SMSVSAM address space in the same lock structure. That might cause differing workloads to interfere with each other and affect system and application availability. You can use secondary lock structures to help prevent locking constraints and allow isolation of workloads.

Follow these steps to implement VSAM RLS multiple lock table:

1. Define the names, sizes and preference locations of the new lock tables in the CFRM policy using the IXM utility.  
-----
2. Activate the new CFRM policy.  
-----
3. Define the names of the new lock tables in the SMS control data set (CDS) using the ISMF Control Data Set function or Navquest. Specify the name of a lock set and one lock structure per lock set (you can use the same lock structure for multiple lock sets). If you do not specify the names of the new lock sets, RLS processing uses the primary lock structure, IGWLOCK00, for all record locking.  
-----
4. Use ISMF or Navquest to add the lock set name to each storage class for which you requested multiple lock table. Otherwise RLS processing uses IGWLOCK00 for all record locking.  
-----
5. Validate and activate the SMS configuration to allow the new definitions in the Source Control Data Set (SCDS) to take effect.  
-----
6. Allocate VSAM data sets using the storage classes that contain the new lock set names.  
-----
7. Update ACS routines.

**Note:**

1. Secondary lock structures are used for record locks only. When secondary lock structures are used, IGWLOCK00 is still required for other types of locks. IGWLOCK00 is required for the initialization of the SMSVSAM address space.
2. The number of lock structures that are concurrently connected varies and depends on the value of MAXCAD specified in the IEASYxx member that is used for system initialization. There might be 10-14 lock structures, including IGWLOCK00, connected at the same time.

For more information, refer to:

- [“Defining the primary CF lock structure” on page 244,](#)
- [“Quiescing or enabling a secondary lock structure” on page 264,](#)
- [“Deleting a VSAM RLS lock structure” on page 264,](#)
- [“Displaying information about a secondary lock structure” on page 264.](#)



---

## Chapter 10. Defining data classes

SMS simplifies data set allocations by introducing data classes. This topic 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. Specifying the Override Space attribute causes the space attributes from the data class to override the space information specified explicitly in the JCL.

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. For example, when you assign a data class with attributes for a VSAM data set organization when defining a non-VSAM data set, you may encounter unexpected results as SMS ignores attributes that do not pertain. 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. To apply your changes to existing data sets, you need to delete and redefine the 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 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 because 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

---

Perform the following steps to define data class attributes.

1. Select option 4, Data Class, from the ISMF Primary Option Menu for storage administrators. This displays the Data Class Application Selection panel.

2. Supply values on the Data Class Application Selection panel:

### **CDS Name**

This must be the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active'. This represents the currently active configuration, but you cannot define or alter the storage groups for the active configuration.

### **Data Class Name**

This is the name of the data class. ISMF primes the field with the last used name.

### **Option**

Select option 3, Define. This displays the Data Class Define panel.

3. Supply values on the Data Class Define panel. 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. Use the DOWN command to display the next page of the panel.

For more information, see:

- [“Defining record and space attributes for data class” on page 118](#)
- [“Defining volume and data set attributes for data class” on page 121](#)
- [“Specifying attributes to handle space constraints during allocation” on page 126](#)
- [“Defining VSAM attributes and specifying media types for data class” on page 128](#)
- [“Defining the encryption management mechanism” on page 131](#)
- [“Defining Shareoptions and RLS attributes for data class” on page 131](#)
- [“Specifying attributes for data set reuse and loading” on page 133](#)
- [“Specifying attributes for backup-while-open \(BWO\) and recovery” on page 133](#)

## Defining record and space attributes for data class

On the Data Class Define panel, the 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.

All of the input 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:



**Description**

Is an optional description for the data class. It can be up to 120 characters.

**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 CIsze value is 4096.

**Override Space**

Specifies whether DATA CLASS attributes override attributes obtained from other sources (JCL, AMS control cards or LIKE=). Override Space flag has two values:

**YES**

data class override takes effect.

**NO**

no data class override. NO is the default.

With YES, the following JCL Space subparameters, including dynamic allocations, such as TSO ALLOCATE, are overridden:

- Space type (CYL,TRK, Block length, or record length plus AVREC)
- Primary Quantity
- Secondary Quantity
- Directory blocks

With YES, the following IDCAMS DEFINE space parameters are overridden:

- Cylinders (Primary, Secondary)
- Tracks (Primary, secondary)

- Kilobytes (Primary, Secondary)
- Megabytes (Primary, Secondary)
- Records (Primary, Secondary)
- Controlintervalsize (CISIZE DATA)
- Freespace (CI-percent, CA-percent)

**Note:**

1. You must explicitly specify Avgrec, Avg Value, Primary, Secondary, and Directory. ISMF primes these fields with blanks, and returns an error message for each blank field. 0 is a valid value.
2. If you set Recorg to KS, ES, RR, or LS, then you must also explicitly specify CIsze data. If you set Recorg to KS, then you must also explicitly specify %Freespace CI and %Freespace CA. ISMF primes these fields with blanks, and returns an error message for each blank field. 0 is a valid value. If Recorg is blank and Override Space is YES, then CIsze is defaulted to 1 unless overridden to another value between 1 to 32768 in the Dataclas definition.
3. SMS gets primary space, secondary space, allocation units, and directory blocks either from JCL or from the data class. The space information in the data class must be all inclusive, otherwise jobs might fail or produce unexpected results.
4. If you set Override Space to YES, the space information specified in data class overrides the space information specified on JCL, even for non-SMS managed data sets. The space information in the data class must be all inclusive, otherwise jobs might fail or produce unexpected results.

**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
- CIsze 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 93000) 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 DASD or mountable volumes a data set can span, with the following exceptions. Valid values are 1 through 59.

For striped VSAM data set, the volume count can be overridden by calculations derived from the sustained data rate (SDR) or the dynamic volume count which ever is greater up to the maximum of 59. For non-VSAM striped data set, volume count is determined by sustained data rate(SDR), dynamic volume count has no affect and the maximum count is 59. For managed mountable volumes, this value can be overridden by the volume count specified in JCL up to a maximum of 255. Dynamic volume count has no affect to tape data sets.

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 EOVS 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, on which you specify volume and data set attributes.

### 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 (except for key range data sets, temporary data sets and system data sets).

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.

Sequential extended format data sets can be version 1 or 2. The user can specify the version number with the second value of DSNTYPE on the DD statement or dynamic allocation equivalent. The PS\_EXT\_VERSION keyword in the IGDSMSxx member in PARMLIB can change the default for the version number.

**Note:** Encrypted data sets are created in version 2 format, regardless of user specification, if a key label is present.

If the sequential extended format data set is striped or if it resides only on one volume, then the version number does not matter. If it is not striped and it resides on multiple volumes, then the version number affects whether the system can use FlashCopy. FlashCopy can handle version 2 in that case but not version 1. PTFs are available to allow proper handling on releases earlier than z/OS 2.1.

### **HFS**

Specifies that the data set is a hierarchical file system (HFS) data set.

### **LARGE**

Specifies that the data set is a large format data set. Large format data sets are physical sequential data sets with the ability to grow beyond the limit of 65 535 tracks per volume.

The BLOCKTOKENSIZE=REQUIRE option in the IGDSMSxx member of SYS1.PARMLIB affects what programs can open these data sets. See the description of the IGDSMSxx member in [z/OS MVS Initialization and Tuning Reference](#).

These data sets cannot be written on a system older than z/OS 1.7. These data sets cannot be read on a system older than z/OS 1.7 if they have more than 65535 tracks on each volume.

### **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. This is like coding DSNTYPE=BASIC on the DD statement but you cannot specify BASIC on this panel.

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 with the extended format attribute be allocated using extended addressability. Having extended addressability allows the data set to be allocated or grow beyond the 4 gigabyte (4 GB) size. The extended addressability data set can be a VSAM data set in any record organization but must be allocated as extended format. The only exception is for linear VSAM data sets which can have the extended addressability attribute and be non-SMS managed which means the data class does not have the extended format attribute set. To allow a VSAM LDS to be allocated or grow beyond 4 GB and be non-SMS managed, the following data class values are required:

- *Data Set Name Type* is blank.
- The *If Ext* value is blank.
- *Extended Addressability* value is Y.
- *Recorg* value is LS.

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:** You can use JCL to override the buffering algorithms that are determined by VSAM.

The available values are:

Keyword	Action
User (the default)	System-managed buffer is not used.
S0	System-managed buffering with sequential optimization is used.
SW	System-managed buffering weighted for sequential processing is used.
D0	System-managed buffering with direct optimization is used.
DW	System-managed buffering weighted for direct optimization is used.
System	<p>Specifies that VSAM chooses how many buffers to use and how they are processed.</p> <p>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).</p> <p>The number of buffers and the buffering technique are determined by the system or the user:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• The user, as specified on the JCL DD card (AMP=('ACCBias=)).</li> </ul> <p>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=.</p>

### Force System Determined Blocksize

If the data class or the user specify DSORG=PS or PO for a new data set, then the system attempts to calculate an optimal value for the maximum block size (BLKSIZE). Normally, the user can override this by coding the BLKSIZE keyword on the DD statement or dynamic allocation equivalent. This might be because the user has a special requirement for an inefficient BLKSIZE value. To specify whether the

system ignores the user-specified maximum block size and calculate the data set's maximum block size when allocating space for the new data set, select one of the following options:

**Y**

(1) If the user codes the BLKSIZE keyword and unallocates the new data set before opening it for writing, then the system discards the BLKSIZE value that the user coded. This implies that the storage administrator knows that the user specified an inefficient value for BLKSIZE. The data set label retains the system-determined value until a program opens it for writing. When a program eventually opens the data set for writing, that program can use or override the system-determined block size.

(2) If a program opens the new data set for writing while the DISP value is NEW or the data set is new with DISP=MOD, then this data class option has no effect and OPEN use the BLKSIZE value that the user coded on the DD statement or dynamic allocation. In this case, the BLKSIZE value that the user specified overrides the value that the system determined. This implies that the user had a reason to override the system-determined value for BLKSIZE.

**N**

The user can override normal system-determined block size logic when the data set is allocated and when it is opened for writing. N (No) is the default.

**RMODE31**

RMODE31 allows you to specify whether to allocate the buffers and control blocks in 31-bit addressable storage. This option applies only to VSAM data sets. For other kinds of data sets, the DCBE macro provides a similar option for assembly language programs.

The values you might specify for RMODE31 are:

**ALL**

Control blocks and buffers above the line.

**BUFF**

Buffers (only) above the line.

**CB**

Control blocks (only) above the line.

**NONE**

Control blocks and buffers below the line.

**Dynamic Volume Count**

Dynamic Volume Count is used during allocation processing to determine the maximum number of volumes a data set can span. The number can be in the range 1 through 59. The default value is 1. 59 is the z/OS volume limit.

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 data set allocation, it specifies the number of TIOT and JFCB(X) entries that are created, which allows for additional volumes to be added dynamically as needed during the lifetime of the data set. For VSAM data sets, the Dynamic Volume count is the maximum number of volumes that all components in the sphere being allocated can span.

Use caution when defining a data class with a large Dynamic Volume Count value. Allocating data sets with a large Dynamic Volume Count value require the same amount of TIOT and JFCB space as an equivalent data set with that number of volumes. This can reduce the number of data sets that can be concurrently allocated by a job. This should be taken into consideration for data sets used by jobs that allocate many data sets concurrently. This also should be taken into consideration for data sets that are allocated to the master scheduler address space, such as data sets that are used in the LNKST concatenation in PROGxx processing. The master scheduler address space has a TIOT size of 12K, which allows for approximately 600 single-volume data sets to be concurrently allocated. Alternatively, if every data set allocated by the master scheduler address space used a Dynamic Volume Count of 59, fewer than 50 data sets could be concurrently allocated. For further information regarding TIOT space calculations, refer to the description of the TIOT SIZE parameter in [ALLOCxx in z/OS MVS Initialization and Tuning Reference](#).

**Note:** Dynamic volume count support when a data set extends is a function of access method end-of-volume (EOV) processing and not the data set type. For products that do not use the standard IBM access method EOV interface for their data sets, DVC might not be supported. Consult with your product representative to see if it supports dynamic volume count.

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.

Dynamic Volume Count is not supported for the following:

- Non-SMS managed data sets or data sets with no associated data class
- System data sets:
  - Page data sets
  - Single-volume data sets such as a catalog BCS or VVDS, VSAM temporary data sets, or any system data set with ACB SDS=ON
- SAM striped data sets
- VSAM data sets with the IMBED or KEYRANGE options
- VSAM EOV Snapshot processing

The Dynamic Volume Count field has significance only when it is larger than administrator-specified data class Volume Count, and any user-specified volume information provided by JCL, IDCAMS, or TSO.

### Compaction

Specifies whether the data set is to be compressed format. You can compress data that is being written to an extended format data set or to a PDSE. This field is ignored for DASD data sets if the data set name type is not EXT or LIBRARY. You can compress data when it is being written to tape. A compressed data set cannot reside on the same tape cartridge as a data set that is not compressed.

The possible values are:

#### Y

Extended format or PDSE data sets are compressed and tape volumes are compacted. The type of DASD compression depends on the COMPRESS option in parmlib member IGDSMSxx. Tape volumes are compacted unless compaction is overridden by the user through JCL/dynamic allocation.

#### N

Data sets are not compressed. Tape volumes are not compacted, unless compaction is requested by the user through JCL/dynamic allocation.

#### T

Extended format data sets are compressed using tailored dictionaries. This overrides the COMPRESS option in parmlib member IGDSMSxx.

#### G

Extended format data sets are compressed using generic dictionaries. This overrides the COMPRESS option in parmlib member IGDSMSxx.

#### ZR

Indicates "zEDC Required", meaning that the system should fail the allocation request if the zEDC function is not supported by the system, or if the minimum allocation amount requirement is not met.

#### ZP

Indicates "zEDC Preferred", meaning that the system should not fail the allocation request, but rather create either a tailored compressed data set if the zEDC function is not supported by the system, or a non-compressed extended format data set if the minimum allocation amount requirement is not met.

**blank**

Data sets are not compressed. Tape volumes might be compacted depending on what was specified by the user on JCL/dynamic allocation, the installation with the COMPACT option in parmlib member DEVSUPxx, or the allocated hardware model. This is the default.

T, G, ZR, and ZP override the compression option set in the IGDSMSxx PARMLIB member, and let you select the type of compression on a data set level. Current users of generic compression can move to using tailored or zEDC compression one data set at a time, as new data sets are created.

For an interactive starting point for Integrated Accelerator for zEDC, and access to a variety of technical resources, see [Integrated Accelerator for zEDC \(www.ibm.com/support/z-content-solutions/compression/\)](http://www.ibm.com/support/z-content-solutions/compression/).

**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.

**System Managed Buffering**

Specifies the amount of virtual storage for SMB Direct Access Bias obtained for buffers when opening the data set. The possible values are:

- 1K to 2048000K
- 1M to 2048M

**EATTR**

A data set level attribute specifying whether a data set can have extended attributes (format 8 and 9 DSCBs) and optionally reside in EAS.

**NO**

No extended attributes. The data set cannot have extended attributes (format 8 and 9 DSCBs) and cannot reside in EAS. This is the default behavior for non-VSAM data sets.

**OPT**

Extended attributes are optional. The data set can have extended attributes (format 8 and 9 DSCBs) and can optionally reside in EAS. This is the default behavior for VSAM data sets.

## Specifying attributes to handle space constraints during allocation

You can specify attributes on Page 2 of the Data Class Define panel to indicate whether 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 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.

As of z/OS V2R2, secondary space reduction (SSR) support is available. With secondary space reduction, Space Constraint Relief applies to both primary and secondary allocations. This feature must be enabled before a retry is attempted and is disabled by default. The space constraint relief is applicable during extend to the current volume and for create or extend to a new volume.

**Note:** The caller of the allocation request can override the Space Constraint Relief parameters in the Data Class. DFSMSdss is an example of a product that might do this. DFSMSdss enables Space Constraint Relief to remove the 5-extent limit without reducing the primary quantity, and in some cases requests the best fit methodology.

For more information about the secondary space reduction support, see the following documentation:

- IBM Redbooks® section on space constraint relief for secondary allocation in *IBM z/OS V2R2: Storage Management and Utilities* ([www.redbooks.ibm.com/abstracts/sg248289.html?Open](http://www.redbooks.ibm.com/abstracts/sg248289.html?Open))
- [Allocation of data sets with the space constraint relief attributes in z/OS DFSMS Using Data Sets](#)

Specify one of the following:

**Y**

Specifies that SMS retry the allocation.

**N**

Specifies that SMS does not retry the allocation if the original amount cannot be acquired.

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 dynamic volume count is also specified in the data class, during space constraint relief phase, SMS updates the volume count with the dynamic volume count when the dynamic volume count is greater.

- If the volume count is 1, 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:
  1. SMS uses a best-fit volume selection method to spread the primary quantity over more than one volume (up to the volume count).
  2. 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 maximum number of extents per volume and the maximum number of volumes per data set vary depending on data set type as follows:

- A basic-format or large-format sequential data set and a direct data set can have up to 16 extents per volume and up to 59 volumes.

- An extended-format sequential data set can have up to 123 extents per volume and up to 59 volumes. Either all or none of these volumes can be arranged into stripes for parallel processing.
- A non-system-managed VSAM data set can have up to 255 extents per component and up to 59 volumes.
- A system-managed VSAM data set can have up to 255 extents per stripe and up to 59 volumes. This extent limit can be removed if the associated data class has extent constraint removal specified. Up to 16 volumes at a time can be read or written in parallel due to striping.
- A PDS can have up to 16 extents and only one volume.
- A PDSE can have up to 123 extents and only one volume.
- An HFS data set can have up to 123 extents per volume and up to 59 volumes.

### **Reduce Space Up to (%)**

Specifies the largest amount by which you want to reduce the requested space quantity when the allocation is retried. SMS attempts to allocate the largest possible space that satisfies the percentage. Valid values are 0 to 99, for both the primary allocation amount and secondary allocation amount.

You must also specify Y for the Space Constraint Relief attribute.

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 that your application cannot tolerate a reduction in the space to be allocated, so only the 5 extent limit is relieved.

For VSAM data sets the reduced amount is a multiple of the CA size.

### **Guaranteed Space Reduction**

Specifies whether or not space reduction on guaranteed space allocations is permitted. Specify one of the following:

**Y**

Specifies that space reduction on guaranteed space allocations is permitted.

**N**

Specifies that space reduction on guaranteed space allocations is not permitted.

This is the default.

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. However, if a specific amount of space is required, then increase your requested space amount to ensure the required amount is allocated.
- 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.
- Extent constraint relief is not enabled and system managed VSAM clusters exceed the 255 total extent limit.

## **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, on which you specify VSAM attributes and media.

## Recorg

Specifies the data set organization, and it resembles the REORG 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 8 on page 129 summarizes which data class attributes apply to each record organization (REORG).

Table 8. Applying Data Class Attributes to Record Organization (Recorg)

Attribute	Record Organization (REORG)				
	Blank	KS	ES	RR	LS
<b>LRECL</b>	x	x	x	x	
<b>RECFM</b>	x				
<b>KEYLEN</b>	x	x			
<b>KEYOFF</b>	x	x			
<b>VOLUME COUNT</b>	x	x	x	x	x
<b>SPACE</b>	x	x	x	x	x
<b>CISIZE</b>	x	x	x	x	x
<b>FREESPACE</b>	x	x			
<b>SHAREOPTIONS</b>	x	x	x	x	x
<b>Retpd or Expdt</b>	x	x	x	x	x
<b>DSNTYPE</b>	x	x			
<b>COMPACTION</b>	x	x			

See the following for more information:

- [z/OS DFSMS Access Method Services Commands](#)
- [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.

## CIsze 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 each control interval and control area to be set aside as free space when the cluster is initially loaded or when a mass insert is done. 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. The valid media types are MEDIA1, MEDIA2, MEDIA3, MEDIA4, MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, MEDIA10, MEDIA11, MEDIA12, MEDIA13, 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.
- If MEDIA5, MEDIA6, MEDIA7, or MEDIA8 is specified, EFMT1, EFMT2, EFMT3 and EEFMT3 are valid.
- If MEDIA9 or MEDIA10 is specified, EFMT2, EFMT3, EFMT4, EEFMT3, and EEFMT4 are valid.
- If MEDIA11, MEDIA12, or MEDIA13 is specified, EFMT4 and EEFMT4 are valid.

Refer to *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.

**Performance Scaling**

Specifies the performance scaling preference. The valid values are Y, N, or blank. This field is optional.

- If you select a value of Y, it means that you want performance scaling. Performance scaling, however, is valid only when you select MEDIA5, MEDIA9, or MEDIA11. This value is not valid if you select any other media type. Also, the appropriate recording technology must be used for the selected media type.
- If you select a value of N or blank, it means that you do not want performance scaling.

Performance scaling and performance segmentation are mutually exclusive.

**Performance Segmentation**

Specifies whether the system should enable segmentation format of the tape. The value specified in this field is considered only for the devices that support the segmentation for performance feature. The valid values are Y, N, or blank. This field is optional.

- If you select a value of Y, it means that you want performance segmentation. Performance segmentation, however, is valid only when you select MEDIA5, MEDIA9, or MEDIA11. This value is not valid if you select any other media type. Also, the appropriate recording technology must be used for the selected media type.
- If you select a value of N or blank, it means that you do not want performance segmentation.

Performance segmentation and performance scaling are mutually exclusive.

**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 the encryption management mechanism

From Page 3 of the **Data Class Define** panel, use the DOWN command to view Page 4, on which you specify the key labels and encoding mechanism.

### Key Label

Specifies the label for the key encrypting key used by the Encryption Key Manager. The key encrypting key is used to encrypt the data (encryption) key. To specify the key label value, you can use up to 64 characters, with blanks to pad the field on the right. The characters can be alphanumeric, national, or special characters with some additional characters also allowed. It is treated as a free form field on input and validity that is checked by the control unit when the key label is first used and converted from EBCDIC to ASCII. The characters that are specified through ISMF must map to ASCII characters X'20' to '7E'.

### Data Set Key Label

Specifies the label for the encryption key used by the access methods. The access methods use the encryption key to encrypt the data. To specify the data set key label value, you can use up to 64 characters, with blanks to pad the field on the right. The characters can be alphanumeric, national, or special characters with some additional characters allowed. The Data Set Key Label is treated as a free form field on input, and validity checked each time the data set is opened.

### Encoding Mechanism

Specifies how the label for the key encrypting key that is specified by the key label (input) is encoded by the Encryption Key Manager and stored on the tape cartridge.

- **L** = encoded as the specified label
- **H** = encoded as a hash of the public key

## Defining Shareoptions and RLS attributes for data class

From Page 4 of the Data Class Define panel, use the DOWN command to view Page 5, on which you specify Shareoptions and RLS attributes.

### 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.

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.

#### **DIRONLY**

indicates that RLS will not cache the data or index parts of the VSAM data set in the coupling facility cache structure. In this case, RLS will use the XCF cache structure to keep track of data that resides in permanent storage (DASD) and in local storage but data or index CIs are not stored in the cache structure itself.

### **RLS Above the 2-GB Bar**

Specifies whether the SMSVSAM address space can take advantage of 64-bit addressable virtual storage during VSAM RLS buffering.

You can specify the following values:

**Y**

Indicates that VSAM RLS buffers can reside above the 2-gigabyte bar. This setting is recommended for best performance of high-volume applications that use VSAM RLS buffering.

**N**

Indicates that VSAM RLS buffering is limited to storage below the bar. **N** is the default.

### **Extent Constraint Removal**

Specifies whether data sets are allowed to be extended beyond 255 extents. See [Using Vsam extents in z/OS DFSMS Using Data Sets](#) for details.

### CA Reclaim

Specifies whether the DASD space for empty control areas (CAs) will be reclaimed for key-sequenced data sets (KSDS). Specify one of the following:

**Y**

Indicates that DASD space for empty CAs will be reclaimed. This is the default.

**N**

Indicates that DASD space for empty CAs will not be reclaimed.

The value for CA Reclaim is always saved in the catalog, but is used only if the CA reclaim function is enabled in the IGDSMSxx member of PARMLIB or with the SETSMS command.

The setting for CA Reclaim for a data class is processed only when a catalog entry for a KSDS is defined. If you need to change the CA reclaim setting for an existing KSDS based on the contents of the data class, either use the ALTER command for the KSDS or delete and define the KSDS to use the value in the data class.

For more information, see the topic about [Reclaiming CA Space for a KSDS in z/OS DFSMS Using Data Sets](#).

## 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. 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:

**ALL**

Tells VSAM to do both forward and backward recovery logging. Changes made by applications are written to the MVS log stream indicated on the logstream ID parameter.

**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**

Tells VSAM to do forward recovery logging. Changes made by applications are written to the MVS log stream indicated on the logstream ID parameter.

**UNDO**

Tells VSAM to do backward recovery logging. 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(ALL), FRlog(REDO), or FRlog(UNDO), you must specify the logstream ID parameter for the VSAM data sets. 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.

### Log Replicate

Specifies whether the data set is eligible for replication:

**Y**

Yes, data set is eligible for VSAM replication.

**N**

No, data set is not eligible for VSAM replication. This is the default.

## 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 RECOG 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, because 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.

For more information on using the LIKE and REFDD keywords, refer to [z/OS MVS JCL Reference](#). For information on determining data classes through ACS routines, see [Chapter 13, “Defining ACS routines,”](#) on page 149. For information on access method services DEFINE defaults and data class defaults, refer to [z/OS DFSMS Access Method Services Commands](#).

## 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 components”](#) on page 208.

---

## Chapter 11. 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 topic 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 DFSMSshsm 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 DFSMSshsm Storage Administration* 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 ISMF. 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.

## 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.

This topic describes defining aggregate groups. See the following for further information on the other options:

- [“Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF” on page 200](#) for additional information on option 1
- [“Altering aggregate groups” on page 207](#) for additional information on option 4
- [“Backing up and recovering an aggregate group” on page 140](#) and [z/OS DFSMSshm Storage Administration](#) for additional information on options 5 and 6.

Perform the following steps to define storage groups.

1. Select option 9, Aggregate Group, from the ISMF Primary Option Menu for storage administrators. This displays the Aggregate Group Application Selection panel.

- 
2. Supply values on the Aggregate Group Application Selection panel:

### **CDS Name**

This is the name of an SCDS. ISMF primes the CDS Name field with the name last used for an aggregate group. The default is 'Active'. This represents the currently active configuration, but you cannot define or alter aggregate groups to the 'Active' configuration.

### **Aggregate Group Name**

This is the name of the aggregate group. ISMF primes the field with the name last used.

### **Option**

Select option 3, Define. This displays the Aggregate Group Define panel

- 
3. Supply values on the Aggregate Group Define panel. You can leave the Aggregate Group Define panel at any time without saving the aggregate group attributes or changes by issuing the Cancel command.

For more information, see:

- [“Defining aggregate group attributes” on page 139](#)
- [“Editing aggregate group attributes” on page 139](#)

## Defining aggregate group attributes

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.

### Description

Describes the aggregate group, and can be up to 120 characters. It is optional.

### Number of Copies

Specifies the number of aggregate backup output files to be created. The valid values are 1-15. It is required.

### 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. It is required.

## Editing aggregate group attributes

After specifying backup attribute values, issue the DOWN command to view page two of the Aggregate Group Define panel. 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 9 on page 140](#). See [z/OS ISPF User's Guide Vol 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 quotation marks. If you do not enclose the name in single quotation marks, 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 quotation marks. If you do not enclose the name in single quotation marks, 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 9 on page 140](#).

```

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 9. 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](#) 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 ALLOCATE.

## **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 components” on page 208](#).





# Chapter 12. Defining copy pools

Copy Pool Name specifies the name of the copy pool. The maximum length is 23 alphanumeric or national characters, or any combination. With one exception where the underscore character ('\_') is also allowed in the combination. The first character can not be numeric or underscore.

DFSMSHsm manages the use of volume-level fast replication functions, such as FlashCopy and SnapShot. These functions provide point-in-time copy services that can quickly copy data from a source location to a target location. Using a copy pool, you can specify the pool storage groups that you want DFSMSHsm to process collectively for fast replication.

Each pool storage group in a copy pool contains the name of the associated target *copy pool backup storage group*. A copy pool backup storage group is a type of SMS storage group that contains eligible target volumes that DFSMSHsm can select for fast replication backup versions. [Figure 10 on page 143](#) shows this relationship.

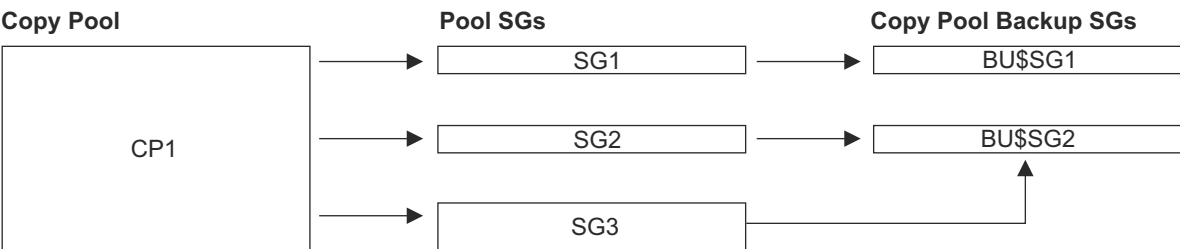


Figure 10. Pool Storage Groups and Copy Pool Backup Storage Groups

After you define a copy pool and a copy pool backup storage group, you can use DFSMSHsm commands to prepare, create, recover, and delete fast replication backup versions. See [Chapter 4, “Defining storage groups,” on page 25](#) to learn how to define a copy pool backup storage group.

This topic shows you how to define a copy pool.

## Before you begin:

1. For information about using DFSMSHsm commands to perform fast replication functions, see [z/OS DFSMSHsm Storage Administration](#).
2. For information about how DFSMSdss supports DFSMSHsm fast replication functions, see [z/OS DFSMSdss Storage Administration](#).
3. For information about FlashCopy and SnapShot, see [z/OS DFSMS Advanced Copy Services](#).

The following table shows the subtasks and associated procedures you perform in order to use DFSMSHsm fast replication commands with copy pools.

Subtask	Associated procedure (See . . . )
Creating copy pools	<ul style="list-style-type: none"><li>• <a href="#">“Planning a copy pool” on page 143</a></li><li>• <a href="#">“Defining a copy pool” on page 144</a></li></ul>
Creating copy pool backup storage groups	<a href="#">“Defining a copy pool backup storage group” on page 45</a>

## Planning a copy pool

Identify the storage groups for which you want DFSMSHsm to manage the fast replication backups. Then determine which of those storage groups you want to put into the same copy pool. Consider these recommendations:

- If a single storage group contains data that you want to assign to separate copy pools, you must place the data into separate storage groups.
- DFSMS does not verify that the storage groups listed in the copy pool include any associated extend and overflow storage groups. Therefore, if you have implemented extend or overflow storage groups, ensure you include them in the appropriate copy pools.
- A pool storage group defined to a copy pool must contain FlashCopy volumes, SnapShot volumes, or any combination of the two.
- Determine whether auto dump should be enabled for any of the copy pools. If so, you must determine:
  - Which host, if any, the copy pool has been defined on (auto dump affinity)
  - Whether to adjust the auto dump window to allow for the new workload
  - Whether any storage groups in a copy pool should be processed by auto dump separately from the copy pool. In this case, you might need to define new dump classes to better separate copy pool dumps from non-copy pool dumps.

For DFSMSshm fast replication recovery processing to recover a deleted or moved data set, user catalogs must exist on volumes that are defined in storage groups that are defined in the copy pool definition. In addition, give careful consideration to the ALLOWPPRC settings in the copy pool. For more information, see [z/OS DFSMSshm Storage Administration](#).

## Defining a copy pool

---

You can use ISMF to define, alter, list, or display copy pool attributes. This topic describes the ISMF panels that you use to define copy pool attributes, and outlines the steps you follow to define a copy pool.

### Steps for defining a copy pool

#### Before you begin:

1. Read the information in [“Planning a copy pool”](#) on page 143.
2. Back up the SMS Source Control Data Set (SCDS).

Perform the following steps to define a copy pool.

1. Select option P, Copy Pool, on the ISMF Primary Option Menu for Storage Administrators. This displays the Copy Pool Application Selection panel.

- 
2. Supply values on the Copy Pool Application Selection panel:

#### CDS Name

Specifies the name of the SCDS where the copy pool is located. Valid values include:

- A data set name that follows TSO naming conventions
- The quoted word 'ACTIVE', which specifies the currently active configuration

The CDS Name field is required. There is no default.

#### Copy Pool Name

Specifies the name of the copy pool. The maximum length is 23 alphanumeric or special characters. The first character cannot be numeric.

The Copy Pool Name field is required. There is no default.

**Guideline:** For Db2, use the required Db2 naming convention when you name the copy pool.

#### Option

Select option 3, Define. This displays the Copy Pool Define panel.

- 
3. Supply values on the Copy Pool Define panel. The SCDS Name and Copy Pool Name fields are primed by ISMF with the values that are specified on the Copy Pool Application Selection panel.

**Description**

Describes the copy pool. You can use up to 120 characters.

Description is optional.

**Auto Dump**

Specifies with Y or N whether volumes in this copy pool are to be eligible for automatic dump processing.

Auto Dump is a required field. The default is N.

**Dump Sys/Sys Group Name**

Specifies the 1-8 character name of the system or system group where volumes in this copy pool are to automatically dump to back-up volumes (the auto dump affinity).

Dump Sys/Sys Group Name is an optional field.

**Dump Class**

Specifies the 1-8 character names of up to five dump classes. ISMF does no validity checking of the values you enter in these fields.

Dump Class is an optional field.

**Number of DASD Fast Replication Backup Versions with Background Copy**

Specifies the number of fast replication backup versions of the copy pool that you want to be maintained by DFSMSHsm. Valid values range from 0 to 85. You can leave the field blank. If you specify 0 (zero), DFSMSHsm creates the DASD backup copy with the NOCOPY option.

Number of DASD Fast Replication Backup Versions with Background Copy is an field optional. The default is 2.

Each backup version that you specify requires a unique target volume for each source volume. Target volumes are defined in the copy pool backup storage group. For more information, see [“Defining a copy pool backup storage group” on page 148](#).

**Recommendation:** Specify a minimum of two backup versions.

**FRBACKUP to PPRC Primary Volumes allowed**

Specifies whether DFSMSHsm will target Metro Mirror or Global Mirror primary volumes, if available, for FRBACKUP processing. The values are:

**NO**

Do not use Metro Mirror or Global Mirror primary volumes as FlashCopy target volumes. This is the default.

**PN**

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the target volume is a Metro Mirror primary device, do not consider Preserve Mirror when determining FlashCopy eligibility and when performing fast replication backup.

**PP**

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the target volume is a Metro Mirror primary volume, a Preserve Mirror operation is preferred.

The IBM DS8880 series and newer IBM storage systems do not support the Preserve Mirror Preferred option for Metro Mirror. If you are using one of those storage systems, and you want to preserve your disk mirror or copy, specify PR.

**PR**

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the target volume is a Metro Mirror primary volume, a Preserve Mirror operation is required.

**blank**

Is the same as NO.

For more information about FlashCopy, Preserve Mirror, synchronous PPRC, also known as synchronous Peer-to-Peer Remote Copy (PPRC), and other copy services functions, refer to [z/OS DFSMS Advanced Copy Services](#).

For more information, see [Using Metro Mirror primary volume during fast replication backup](#) in [z/OS DFSMSHsm Storage Administration](#).

#### **FRRECOV to PPRC Primary Volumes allowed**

Specifies whether DFSMSHsm will target Metro Mirror or Global Mirror primary volumes, if available, for FRRECOV processing. The values are:

##### **NO**

Do not use Metro Mirror or Global Mirror primary volumes as FlashCopy target volumes. This is the default.

##### **PN**

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the target volume is a Metro Mirror primary device, do not consider Preserve Mirror when determining Flashcopy eligibility and when performing fast replication backup.

##### **PP**

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the target volume is a Metro Mirror primary volume, a Preserve Mirror operation is preferred.

The IBM DS8880 series and newer IBM storage systems do not support the Preserve Mirror Preferred option for Metro Mirror. If you are using one of those storage systems, and you want to preserve your disk mirror or copy, specify PR.

##### **PR**

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the target volume is a Metro Mirror primary volume, a Preserve Mirror operation is required.

The IBM DS8880 series and newer IBM storage systems do not support the Preserve Mirror Preferred option for Metro Mirror. If you are using one of those storage systems, and you want to preserve your disk mirror or copy, specify PR.

##### **blank**

Is the same as NO.

For more information about FlashCopy, Preserve Mirror, synchronous PPRC, also known as synchronous Peer-to-Peer Remote Copy (PPRC), and other copy services functions, see [z/OS DFSMS Advanced Copy Services](#).

For more information, refer to the topic about [Using Metro Mirror primary volume during fast replication recovery](#) in [z/OS DFSMSHsm Storage Administration](#).

#### **FlashCopy Consistency Group**

Specifies whether DFSMSHsm will use consistency groups for FlashCopy. The values are:

##### **N (No)**

Do not use consistency groups for FlashCopy. This is the default.

##### **Y (Yes)**

Use consistency groups for FlashCopy, to create data-consistent copies.

##### **blank**

Is the same as N (No).

For more information, refer to the topic about [Creating consistent copies using FlashCopy consistency groups](#) in [z/OS DFSMSHsm Storage Administration](#)

#### **FRBACKUP to XRC Primary Volumes allowed**

Specifies whether DFSMSHsm will target XRC primary volumes, if available, for FRBACKUP processing. The values are:

##### **N (No)**

Do not use XRC primary volumes as FlashCopy target volumes. This is the default.

**Y (Yes)**

XRC primary volumes are allowed to become FlashCopy target volumes.

**blank**

Is the same as N (No).

For more information about FlashCopy, XRC, also known as extended remote copy, and other copy services functions, refer to [z/OS DFSMS Advanced Copy Services](#).

**FRRECOV to XRC Primary Volumes allowed**

Specifies whether DFSMSHsm will target XRC primary volumes, if available, for FRRECOV processing. The specified value is considered in selecting FlashCopy target volumes for FRBACKUP processing during volume pairing. However, the LO volumes are not verified explicitly for remote pair FlashCopy eligibility for FRRECOV processing during DFSMSHsm volume pairing. The values are:

**N (No)**

Do not use XRC primary volumes as FlashCopy target volumes. This is the default.

**Y (Yes)**

XRC primary volumes are allowed to become FlashCopy target volumes.

**blank**

Is the same as N (No).

For more information about FlashCopy, XRC, also known as extended remote copy, and other copy services functions, refer to [z/OS DFSMS Advanced Copy Services](#).

4. Use the DOWN command to view page 2 of the panel.

5. Specify the values on page 2:

**Catalog Name**

Specifies the names of one or more valid catalogs (up to 10). Use fully qualified names, without quotation marks. Catalog Name is required if Capture Catalog Information for Data Set Recovery is R or P. There is no default.

**Capture Catalog Information for Data Set Recovery**

Specifies, to FRBACKUP processing, options for collecting catalog information. The options are:

- R (Required). If not able to capture catalog information, fail the backup version.
- P (Preferred). If not able to capture catalog information, issue a warning and do not fail the backup version.
- N (do not collect catalog information).

N is the default.

**Note:** It is not uncommon for mature or highly utilized catalogs to contain logical errors. However, the catalog capture function in DFSMSHsm Fast Replication requires the catalogs to be free of all errors to ensure that each cataloged data set can be recovered. If errors are present in a catalog when capture catalog information has been requested, message ARC1812I, indicating a catalog failure, may be issued. If the Required option was specified, the fast replication backup request will fail. To allow fast replication backup to succeed, you should correct all catalog errors. DFSMSHsm uses the Catalog Search Interface (CSI) to capture the catalog information. Thus, if the errors in a catalog cause CSI to return with an error, DFSMSHsm issues an ARC1812I message. To allow Fast Replication Backup to succeed, you should correct all catalog errors. Alternatively, specify the Preferred option, which will fail the capture catalog function but allow the FRBACKUP request to continue processing.

**Allow Fast Reverse Restore**

Indicates whether to allow recovery of a FlashCopy source from the FlashCopy target without waiting for the background copy to complete. The options are:

- Y (Yes). Allow fast reverse restore.
- N (No). Do not use fast reverse restore. This is the default.

6. Use the DOWN command to view page 3 of the panel.

---

7. Specify the names of one or more valid pool storage groups. You can specify up to 256 valid pool storage group names in the fields that appear on pages 3, 4 and 5 of the Copy Pool Define panel. Pages 4 and 5 of the panel are not shown here. Use the DOWN command to display them.

Storage Group Names is a required field. You must specify at least one storage group name. There is no default.

**Rule:** You must specify all associated extend and overflow storage groups to ensure that they are included in the copy pool.

---

8. Use the END command to save and exit the panel.

**Result:** When you are done, you have defined a new copy pool.

## Defining a copy pool backup storage group

---

After you define a copy pool, you must define a copy pool backup storage group. A copy pool backup storage group contains the target volumes for fast replication backup versions. For information on how to use ISMF to define a copy pool backup storage group, see [Chapter 4, “Defining storage groups,” on page 25](#).

Each copy pool backup storage group must be associated with a pool storage group. You must specify the name of the copy pool backup storage group on the Pool Storage Group Define/Alter panel.

You can use the BACKUPSTORAGEGROUP parameter to override the copy pool backup storage group with an alternate storage group. You can specify this optional keyword with the EXECUTE and PREPARE keywords of the FRBACKUP command for DFSMSHsm: Requesting a Fast Replication Backup or Dump Version. For more information, see [z/OS DFSMSHsm Storage Administration](#).

---

## Chapter 13. Defining ACS routines

This topic documents intended Programming Interfaces that allow you to write programs to obtain the services of DFSMS.

This topic 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.

---

### 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.

When a data set is allocated, it is assigned an initial management class, storage class, and storage group by defaults defined by an ACS routine or by explicit request. These assignments can be changed through the DFSMSshm class transition processing that occurs during the automatic space management functions. During the automatic class transition, ACS routines are invoked with the ACS environment of SPMGCLTR to determine the new management class, storage class, and storage group assignments.

Chapter 19, “Writing ACS routines,” on page 267 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 [Chapter 24, “Using NaviQuest,”](#) on page 323 for 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
- DFSMSshm RECALL, RECOVER, and class transitions
- 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.

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 class 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 that neither the remote user nor the data class specified, the creation process fails.

See [“Determining Distributed FileManager/MVS data set creation requests”](#) on page 288 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 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
- VVDS data sets: SYS1.VVDS.\*
- VTOC Index data sets: SYS1.VTOCIX.\*

## Creating ACS routines

**Before you begin:** 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.

Perform the following steps to create ACS routines.

1. Select option 7, Automatic Class Selection, from the ISMF Primary Option Menu. This displays the ACS Application Selection panel.

2. Select option 1, Edit, on the ACS Application Selection panel. This displays the Edit Entry panel.



3. Supply the required values on the Edit Entry panel, then press Enter to invoke the ISPF Editor.

-----  
4. Supply the source code for the ACS routine.

-----  
5. Use the END command to save the ACS routine and return to the ISPF Edit Entry panel.

-----  
6. Use the END command again to return to the ACS 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.

**Before you begin:** You must display the ACS Application Selection panel. See [“Creating ACS routines”](#) on page 150.

Perform the following steps to translate ACS routines.

1. Select option 2, Translate, from the ACS Application Selection panel. This displays the Translate ACS Routines panel.

-----  
2. Supply values on the Translate ACS Routines panel:

**SCDS Name**

Is the name of an SCDS. A successful translation places the ACS routine object table into this SCDS. SCDS Name is primed with the last SCDS that you have referenced or translated into.

**ACS Source Data Set**

Is the name of the data set containing the ACS routine that you want to translate. 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. It 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.

**Listing Data Set**

Is 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.

## Browsing the results of a translation

If you specify an ACS Source Data Set that contains no errors, you see a PDF browse panel after 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 *z/OS MVS System Messages, Volumes 1–10* 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 see the Output Listing Disposition panel. The short message area displays the results of the translation.

Supply these values on the Output Listing Disposition panel:

**Print Output Listing**

Specify Y (Yes) to submit a batch job to print the data set. The default is N (No).

**Delete Output Listing**

Specify Y (Yes) to delete the data set after it is printed. The default is N (No).

Table 9 on [page 152](#) summarizes the possible outcomes for each combination of Output Listing Disposition values.

<i>Table 9. Results of Output Listing Disposition</i>			
<b>Print</b>	<b>Delete</b>	<b>Resulting Action</b>	<b>Type of Job</b>
<b>Yes</b>	Yes	Print data set DISP=(OLD,DELETE)	Batch
<b>Yes</b>	No	Print data set DISP=(OLD,KEEP)	Batch
<b>No</b>	Yes	Delete data set	Foreground
<b>No</b>	No	No action (dataset kept)	N/A

Use 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 11 on page 153](#) 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 07/15/2004 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 > 9999999KB)
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 11. 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

**Before you begin:** Display the ACS Application Selection panel. See [“Creating ACS routines” on page 150](#).

Perform the following steps to validate an ACS routine.

1. Select option 3, Validate, on the ACS Application Selection panel. This displays the Validate ACS Routines or Entire SCDS panel.

2. Supply values on the Validate ACS Routines or Entire SCDS panel:

#### 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.

---

3. Press Enter to perform the validation. See [“Validation results” on page 154](#) for more information.

## Validation results

Validation fails if certain conditions are not satisfied, as described below (except when, as noted, only a warning is issued).

- **For the storage group ACS routine**, the conditions are:

- 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.

**Note:**

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'
```

## Browsing validation results

If you specified a data set name in the Listing Data Set field, you get the Browse panel shown in [Figure 12 on page 155](#) 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:  2004/11/19
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 12. 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 *z/OS MVS System Messages, Volumes 1–10* for an explanation of the diagnostic messages.

When you leave Browse, you receive the Output Listing Disposition panel. See [“Browsing the results of a translation”](#) on page 151 and [Table 9](#) on page 152 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 IGDSMSxx. If the SCDS is not valid, activation fails.

**Before you begin:** Display the CDS Application Selection panel (select option 8 from the ISMF Primary Option Menu for Storage Administrators.)

Perform the following steps to validate an entire SCDS.

1. On the command line of the CDS Application Selection panel, type the VALIDATE command. This displays the Validate ACS Routines or Entire SCDS panel.

- 
2. Supply values on the Validate ACS Routines or Entire SCDS panel:

### SCDS Name

Identifies the data set you want to validate. It is a required field that is primed with the last used value. The default is blank.

### ACS Routine Type

Type an \* in this field, because you are validating an entire SCDS.

### 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.

- 
3. Press Enter to perform the validation. See [“Validation results”](#) on page 154 for more information.

## Validation results

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 208](#) 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 that 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

Because 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, because 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

**Before you begin:** 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. Then, display the ACS Application Selection panel. See [“Creating ACS routines” on page 150](#).

Perform the following steps to create ACS test cases.

1. Select option 4, Test, from the ACS Application Selection panel. This displays the ACS Test Selection panel.

- 
2. Supply values on the ACS Test Selection panel:

**ACS Test Library**

**ACS Test Member**

These identify the data set and member that contain the test case. You can specify one test case per member.

**Option**

Select option 1, Define, to specify test criteria. This displays the ACS Test Case Define panel.

- 
3. Supply values on the ACS Test Case Define panel, which has multiple pages. On these pages, you can specify test values that correspond to the ACS variables listed in [Chapter 19, “Writing ACS routines,” on page 267](#).

On page 3, you can specify values corresponding to the MSxxxx parameters that you expect to be returned from your tape management system through the pre-ACS routine. See [“Tape management](#)

system support” on page 168 for a description of these parameters, and use the documentation provided by your tape management vendor for more specifics.

- 
4. Use the END command to save your test values and return to the ACS Test Selection panel. On the ACS 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

Perform the following steps to run ACS test cases.

1. Display the ACS Test Selection panel, as described in [“Creating ACS test cases” on page 157](#).

- 
2. Supply values on the ACS Test Selection panel:

**Option**

Select option 3, Test.

**ACS Test Library**

Supply the library name.

Press Enter to see the Test ACS Routines panel.

- 
3. Supply values on the Test ACS Routines panel to indicate which routines you want to test.

**ACS Test Library and Member**

These identify the data set and member that contain the test case. Specify an asterisk in the Member field to run all test cases in the partitioned data set library.

**Listing Data Set**

Specify a data set name to create a listing of the results of the testing. Leave this field blank to prevent the creation of a listing.

- 
4. Press Enter to test the routines. If you specified a listing data set, the results of the testing are displayed in the PDF Browse panel.

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 10 on page 158 shows which ACS routines are invoked when performing initial allocation, importing, copying, restoring and recalling data sets.

Table 10. 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\*](#) 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 11 on page 160 shows which ACS routines are invoked for restoring, recalling, recovering, and converting data sets.

Table 11. 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 Class Transitions	No	Yes	Yes	Yes
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.

Before recalling a data set, DFSMShsm checks if a class transition was missed while that data set was migrated. If a class transition was missed, then DFSMShsm runs the ACS routines with ACS environment SPMGCLTR to determine any new SMS construct names. Using the newly retrieved management class name, DFSMShsm repeats the check to see if a class transition was missed, until the appropriate management and storage classes and storage groups are found. These new management class, storage class, and storage groups are then used for the RECALL.

### Note:

1. To avoid repeating this process, DFSMShsm repeats the check to see if a class transition was missed up to 5 times.
2. DFSMShsm does *not* check if any class transitions were missed during RECOVER, ARECOVER, or FRRECOV of a data set.

## DFSMShsm class transitions

Class transition occurs during the second phase of volume-level space management. Data sets that are eligible for a class transition are processed first and are only transitioned if they are not eligible for

migration as well. This prevents the data from being transitioned to another volume, only to be later migrated.

When a data set is eligible for a class transition, the SMS ACS routines are invoked with an ACS environment of SPMGCLTR.

The data class ACS routine is not invoked.

The storage class ACS routine is invoked. If DFSMSHsm is unable to move the data set to a new storage class, then the data set is not transitioned unless a new storage group has also been assigned. The storage class is derived based on the existing management class and the existing storage group of the data set.

The management class ACS routine is invoked. If the new management class has no transition criteria then no actual data movement is performed. DFSMSHsm assigns the data set to the new management class. The management class is derived based on the existing management class and the new storage class assigned.

The storage group ACS routine is invoked. If a new storage group is assigned, DFSMSHsm attempts to move the data set to the appropriate storage group. The storage group is derived based on the new management class and the new storage class assigned.

## 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 MSPPOOL 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 the referenced data set if the referenced data set has a storage class assigned to it. However, if the referenced data set is an SMS-managed tape data set and it is not cataloged or DISP is not OLD or MOD, the storage class is not available. In that case, the ACS routine is called and must specify a valid storage class for the referencing data set to be an SMS-managed tape data set.

**Note:**

1. Non-SMS managed data sets do not have storage classes to copy. As a result, the storage class routine is invoked and either allows the non-SMS allocation or fails it. If it does anything else, SMS fails it.
2. When VOL=REF is coded to refer to a DD statement and the referenced DD allocates a new data set (DISP=NEW), the storage class is copied from the referenced DD if one is assigned.

When VOL=REF is coded to refer to an SMS-managed tape data set, the storage class from the referenced data set will not be available. The storage class ACS routine is invoked and a valid storage class must be assigned; otherwise, SMS fails it. For example, in the case of VOL=REF to an SMS-managed tape data set, the storage class ACS routine must use the read-only variables &LIBNAME,&ANYVOL,&ALLVOL to derive a valid storage class that will allow this referencing data set to be allocated to SMS-managed tape. Additionally, read-only variable &LABEL can be used to help derive the valid storage class.

3. The storage class ACS routine is also invoked when VOL=REF and UNIT=AFF is coded on the DD statement. For information on what is passed to the storage class ACS routine, refer to [“Non-data set stacking allocations”](#) on page 166.
  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 you specify VOL=REF in the JCL, the system retrieves the volume serial numbers from the referenced DD. In the case of NEW to NEW referencing, since DD is not allocated yet, the default of 1 is passed in the &NVOL parameter.

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 because it has not been allocated yet. In that case only the *first* candidate storage group is passed in.

## Examples: using volume reference

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 13 on page 164](#) 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 13. Example of REF=ST Values when Using VOL=REF*

- 
- 'REF=NS' (the volume references is to a non-SMS-managed data set)

Figure 14 on page 164 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 12/31/04 ALLOCATION WILL BE FAILED FOR ' &DSN
    END
    .
    .
    .
END

```

*Figure 14. Example of REF=NS Values when Using VOL=REF*

---

As of the date shown in Figure 14 on page 164, you might change the ACS routine to fail the uses of VOL=REF that are not valid, as shown in Figure 15 on page 164.

---

```

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 15. 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. Because 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 12 on page 165 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 12. Values for &UNIT ACS Read-Only Variable

&UNIT Value	ACS Invocation	Data Set Stacking Indication	Device Category of Data Set on Which to Stack
AFF=	First	Unknown	Not applicable
STK=SMSD	Second	Yes and different device categories	System-managed DASD

Table 12. Values for &UNIT ACS Read-Only Variable (continued)

&UNIT Value	ACS Invocation	Data Set Stacking Indication	Device Category of Data Set on Which to Stack
<b>STK=NSMS</b>	Second	Yes and different device categories	Non-system-managed DASD or Non-system-managed Tape
<b>STK=SMSD or STK=NSMS</b>	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.

### Examples: storage class ACS routines for read-only variables

Figure 16 on page 166 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 16. 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 17 on page 166:

```
PROC  &STORGRP
SELECT(&UNIT)
  WHEN('STK=SMSD')
    SET &STORGRP = 'S1P03'
  WHEN('STK=NSMS')
    SET &STORGRP = 'POOLSG'
  OTHERWISE
```

Figure 17. 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
<b>DD1</b>	'SYSDA'
<b>DD2</b>	'AFF=SMSD' (if DD1 is directed to SMS DASD)
<b>DD3</b>	'AFF=SMST' (if DD2 is directed to SMS tape)
<b>DD4</b>	'AFF=SMSD' (if DD1 is directed to SMS DASD)



During the C/I phase of processing, SMS IDAX will invoke the ACS routines for every DD statement which specifies DISP=NEW or MOD, including the DD statements having UNIT=AFF=ddname specified. When a storage class is assigned to the AFF'ed DD, it will be passed as an ACS variable, &STORCLAS, to the ACS routines when processing the AFF'ing DD.

If the AFF'ed DD DISP is anything other than NEW, the ACS routines will have to be called again after IDAX by MVS Allocation to finish processing the AFF'ing DD statement. With the exception of the JES3 environment, ACS routines can be called multiple times. When the ACS routines are invoked by JES3 PRESCAN processing, the &UNIT read-only variable are set to AFF= for DD2, DD3, and DD4. See [Table 13 on page 167](#) for more details when the ACS routines are called again to finish processing the AFF'ing DD statement.

[Table 13 on page 167](#) illustrates how &LIBNAME, &STORGRP, &ANYVOL, and &STORCLAS are set when the ACS routines are called again outside of IDAX to finish processing the DD statements containing the UNIT=AFF parameter. 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 of the AFF'ing DD (DD2).

Table 13. AFFing DD Volser Values						
	AFFing DD Volser Values	&LIBNAME	&STORGRP	&ALLVOL	&ANYVOL	&STORCLAS
<b>DD2 volser= V2</b>	V2 = lib resident	libnamesofV2	sgnameofV2	V2	V2	blank
	V2/= lib resident	blank	blank	V2	V2	blank
<b>DD2 volser= blank</b>	DD1= SMSDASD	blank	sgnameofDD1	blank	blank	scnameofDD1
	DD1= nonSMS	blank	blank	blank	blank	blank
	DD1volser =V1	libnameofV1	sgnameofV1	blank	blank	blank
	DD1=SMSTAPE DD1volser =blank DISP=NEW	blank	blank	blank	blank	blank

**Note:** When both the AFFing DD and the AFFed DD are DISP=NEW, the storgrp and the library values will not be available when ACS routines are being processed for the AFFing DD.

## 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: storage class ACS routine specifying both volume reference and unit affinity**

In [Figure 18 on page 168](#), 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 &ANYVOL = 'REF=SD' THEN
  SET &STORCLAS = &STORCLAS
ELSE
  IF &UNIT = 'AFF=SMST' OR &UNIT = 'AFF=SMSD' THEN
    DO
      IF &STORCLAS = '' THEN
        SET &STORCLAS = 'xxxxxxx' /* TO BREAK AFFINITY */
      ELSE
        SET &STORCLAS = &STORCLAS /* TO HONOR AFFINITY */
      END
    END
  WRITE 'STORCLAS SET TO '&STORCLAS

```

*Figure 18. 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 Managing and Using Removable Media* 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.

The STORCLAS ACS routine is not redriven or invoked on data set rename. Redriving the MGMTCLAS ACS routine on data set rename invokes the management class ACS routine when an SMS managed cluster, Generation Data Set (GDS) or non-VSAM data set is renamed. Catalog management invokes the 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. If the data set was ever transitioned before it was migrated or if one or more class transitions was missed while the data set was migrated, and now the data set is being recalled, then the &ACSENVIR field contains SPMGCLTR for all ACS routines except data class ACS routine.

## 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.

An application program may set its own ACS environment for the logical data set COPY and RESTORE commands. The ACS environment may be set when the application is presented with the EIREC22 data area in Exit 22. The specification of an ACS environment will override the default &ACSENVIR values for the logical data set COPY and RESTORE commands. The read only variables available will be no different than the ones available for the COPY command (&ACSENVIR='ALLOC') and the RESTORE command (&ACSENVIR='RECOVER').

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 14 on page 170](#) lists the read-only variables that are set for DFSMSrmm requests.

<i>Table 14. DFSMSrmm Read-Only Variables</i>		
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

Table 14. DFSMSrmm Read-Only Variables (continued)

Variable	RMMPOOL Environment	RMMVRS Environment
<b>&amp;ACSENVIR</b>	Yes RMMPOOL	Yes RMMVRS
<b>&amp;DSN</b>	Yes	Yes
<b>&amp;HLQ</b>	Yes	Yes
<b>&amp;LLQ</b>	Yes	Yes
<b>&amp;NQUAL</b>	Yes	Yes
<b>&amp;ACCT_JOB</b>	Yes	Yes
<b>&amp;ACCT_STEP</b>	Yes	Yes
<b>&amp;FILENUM</b>	Optional for mount messages	Yes
<b>&amp;LABEL</b>	Yes	Yes
<b>&amp;LIBNAME</b>	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='CTTRANS'), 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

---

First, display the ACS Application Selection panel. See [“Creating ACS routines”](#) on page 150.

Then, to display information about the ACS objects stored in a control data set, Select option 5, Display. This displays the ACS Object Display panel, which includes the following 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

---

**Before you begin:** Display the ACS Application Selection panel. See [“Creating ACS routines” on page 150](#).

Perform the following steps to display information about the ACS objects stored in a control data set.

1. Select option 6, Delete, on the ACS Application Selection panel. This displays the Delete ACS Object panel, on which:

**SCDS Name**

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.

**ACS Routine Type**

Contains the type of ACS object. It is a required field and it is primed with the last used value. The default is blank.

2. Press Enter to see the Confirm Delete Request panel to confirm that you want to delete the ACS object.

## Using security labels in ACS routines

---

You can use an automatic class selection (ACS) routine read-only security label variable, &SECLABL, as input to the ACS routine. The value of the security label (&SECLABL) is defined for specific users or data sets.

The security label is a name used to represent the association between a particular security level and a set of security categories. It indicates the minimum level of security required to access a data set protected by this profile. You can set the security label by entering it in the RACF profile of the data set.

**Before you begin:** To understand the concepts of security labels, read [z/OS Planning for Multilevel Security and the Common Criteria](#).

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

Tasks	Procedure that you must perform:
<a href="#">“Planning and installing” on page 174</a>	<ul style="list-style-type: none"> <li>• <a href="#">“Planning to use security labels in ACS routines” on page 174</a></li> <li>• <a href="#">“Installing security labels in ACS routines” on page 174</a></li> </ul>
<a href="#">“Administering” on page 175</a>	<ul style="list-style-type: none"> <li>• <a href="#">“Creating a security label in the RACF data set or user's profile” on page 175</a></li> <li>• <a href="#">“Using JCL or dynamic allocation to extract the security label from the RACF user's profile” on page 176</a></li> <li>• <a href="#">“Specifying the &amp;SECLABL read-only variable in the ACS routine” on page 176</a></li> </ul>

## Planning and installing

This section describes the planning and installation tasks that you must perform to implement security labels in ACS routines.

### Planning to use security labels in ACS routines

**Before you begin:** An installation can define its own security labels. You can use security labels to associate a specific security level with a set of (zero or more) security categories. Consider using security labels if you do not want to mix data for different classified projects. Security labels allow you to separate data of specific classifications on specific sets of volumes. You can assign an appropriate storage group dedicated to data of that classification.

Perform the following steps to help you decide whether and how to use security labels in ACS routines:

1. Decide whether you want to separate data of specific classifications on specific sets of volumes.

-----

2. Create a list of security label names that you can use to separate groups of data.

-----

3. Decide which storage groups should use security labels.

-----

4. Decide which storage group ACS routines to update with the &SECLABL read-only variable.

-----

Now you are ready to implement security labels in ACS routines.

### Installing security labels in ACS routines

Perform the following steps to implement security labels in ACS routines:

1. Create the security label and give users access to it. For more information, see [“Creating a security label in the RACF data set or user's profile” on page 175](#).

-----

2. Specify the DD SECMODEL or DD PROTECT=YES parameter in the JCL or dynamic allocation to extract the security label from the RACF user's profile. Otherwise, the security label is extracted from the data set profile. For more information, see [“Using JCL or dynamic allocation to extract the security label from the RACF user's profile” on page 176](#).

-----

3. Update the storage group ACS routine with the &SECLABL read-only variable.

-----

4. Use the ISMF ACS Test Case Define/Alter application to test the security labels in the storage group ACS routines. For more information, see [“Writing ACS test cases using SECLABEL” on page 178](#).

-----

5. Validate and activate the SCDS.

- a. Select option 8 from the ISMF Primary Option Menu for Storage Administrators, to invoke the Control Data Set (CDS) Application Selection panel on ISMF.
- b. Specify the name of the SCDS to validate in the CDS NAME field.
- c. Select option 4, Validate the SCDS.

-----

Now you are ready to use security labels in storage group ACS routines to separate different types of data.



## Administering

Administration steps include creating security labels, extracting security label values from a RACF profile, and updating ACS routines with the &SECLABL variable. First, you set the **seclabel** value from either the user's RACF profile or from the RACF data set profile as input to the ACS routines. The **seclabel** value is set to null if the RACF SECLABEL class is not active. If you specify the DD SECMODEL or PROTECT=YES parameter in the JCL or dynamic allocation, the SECLABEL is extracted from the RACF data set profile. Your installation's ACS routines can assign storage groups depending on the &SECLABL value along with other read-only ACS variables.

**Tip:** If overflow storage groups or extended storage groups are also defined, ensure that the security levels do not conflict.

**Restriction:** For z/OS V1R6 DFSMS, customers cannot use &SECLABL in ACS routines if they are using automatic data set protection (ADSP).

### Creating a security label in the RACF data set or user's profile

**Before you begin:** To understand security labels, read *z/OS Security Server RACF Security Administrator's Guide*. For information about RACF commands, see *z/OS Security Server RACF Command Language Reference*.

Perform the following steps to create a security label:

1. Define the SECLEVEL profile to the SECDATA class using the RACF RDEFINE command.

Example: RDEFINE SECDATA SECLEVEL UACC(NONE)

- 
2. Define security levels as members of the SECLEVEL profile in the SECDATA class.

Example:

```
RALTER SECDATA SECLEVEL ADDMEM(seclevel-name/seclevel-number ...)
```

- 
3. Define the CATEGORY profile to the SECDATA class using the RDEFINE command.

Example: RDEFINE SECDATA CATEGORY UACC(NONE)

- 
4. Define categories as members of the CATEGORY profile in the SECDATA class.

Example:

```
RALTER SECDATA CATEGORY ADDMEM(category-1 category-2 ...)
```

- 
5. For each security label, define a profile in the SECLABEL class.

Example:

```
RDEFINE SECLABEL security-label SECLEVEL(seclevel-name) ADDCATEGORY(category-1 category-2 ...)
```

- 
6. Provide READ access authority to each user of the security label. In this example, EAGLE is the name of the security label.

Example:

```
PERMIT EAGLE CLASS(SECLABEL) ACCESS(READ) ID(AHLEE GROUP1)
```

---

7. When you are ready to start using security labels, activate the SECLABEL class and activate SETROPTS RACLIST processing for the class.

Example: SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)

---

Now you are ready to assign the security label to the &SECLABL read-only variable to use in the storage group ACS routine.

## Using JCL or dynamic allocation to extract the security label from the RACF user's profile

**Before you begin:** Specify the DD SECMODEL or DD PROTECT=YES parameter in the JCL or dynamic allocation to extract the security label from the RACF user's profile. Otherwise, the security label is extracted from the data set profile. For information about using JCL statements, see [z/OS MVS JCL Reference](#). For information about using dynamic allocation, see [z/OS MVS Programming: Authorized Assembler Services Guide](#).

Perform the following step to extract the security label from the user's profile:

1. Specify the DD SECMODEL or DD PROTECT parameter in the JCL or dynamic allocation, then submit the job.

---

Example using the SECMODEL parameter:

```
//STEP20 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=A
//DD3 DD DSN=USER#1.S16SL001.DATASET3,
// DISP=(NEW,CATLG),SPACE=(TRK,(2,5)),
// STORCLAS=S1P01S01,UNIT=3390,
// SECMODEL=USER#1.S16SL001.MODEL.DATASET
```

The above example specifies the DD SECMODEL parameter in JCL to extract a security label from the discrete profile. The USER#1.S16SL001.MODEL.DATASET data set has been assigned a security label. The security label of the newly allocated data set USER#1.S16SL001.MODEL.DATASET is extracted from the RACF discrete profile of user USER#1.

Example using the PROTECT parameter:

```
//STEP16 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=A
//DD1 DD DSN=USER#1.S16SL002.DATASET1,
// DISP=(NEW,CATLG),SPACE=(TRK,(2,5)),
// STORCLAS=S1P01S01,UNIT=3390,PROTECT=YES
```

The above example specifies the DD PROTECT=YES parameter in JCL to extract a security label from the discrete profile. The security label of the newly allocated data set USER#1.S16SL002.DATASET1 is extracted from the RACF discrete profile of user USER#1.

2. Submit the job.

The security label is extracted from the specified user's profile.

---

**Result:** Now you are ready to assign the security label to the &SECLABL read-only variable to use in the storage group ACS routine.

## Specifying the &SECLABL read-only variable in the ACS routine

**Before you begin:** 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. To understand how to test ACS routines and use read-only variables, read [“Testing ACS routines”](#) on page 157 and Chapter 19, [“Writing ACS routines,”](#) on page 267.

Perform the following steps to use the &SECLABL read-only variable in the ACS routine:

1. Update the storage group ACS routine with the &SECLABL read-only variable. This example assumes that a RACF security label ALERT is already defined to the system.

Example:

```
PROC &STORGRP
SELECT
WHEN (&SECLABL = 'ALERT')
DO
  SET &STORGRP = 'S1P01'
  WRITE 'ASSIGN DATA SETS WITH SECLABEL ALERT TO STORAGE GROUP: S1P01'
  EXIT CODE(0)
END
OTHERWISE
DO
  SET &STORGRP = 'S1P02'
  WRITE 'ASSIGN DATA SETS WITHOUT SECLABEL ALERT TO STORAGE GROUP: S1P02'
  EXIT CODE(0)
END
END
END
```

- 
2. Validate the storage group ACS routine.

- a. On the ISMF Primary Option menu, select option 7, Automatic Class Selection.
- b. On the ACS Application Selection panel, select option 3, Validate to validate the ACS routine.
- c. On the Validate ACS Routines or Entire SCDS panel, specify the name of the SCDS where the ACS routine resides.
- d. Specify the ACS routine type, which is SG (storage group).
- e. Specify the name of the listing data set to where the validation results are printed.
- f. Press Enter to validate the storage group ACS routine. On the CDS Application Selection panel, select the Validate option.

- 
3. Test your storage group ACS routines. For more details, see [“Writing ACS test cases using SECLABEL”](#) on page 178.

- 
4. Validate the SCDS with the updated storage group ACS routine.

- 
5. Activate the validated SCDS.

On the CDS Application Selection panel in ISMF, complete one of the following actions:

- Select the Activate option, or
- Enter the ACTIVATE operator command on the command line.

Example: ACTIVATE SCDS *dsname*

From the operator console, enter the SETSMS command.

Example: SETSMS SCDS *dsname*

---

Now you are ready to use security labels in storage group ACS routines to separate different types of data.

## Writing ACS test cases using SECLABEL

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. To write an ACS test case using the SECLABEL value, follow these steps:

1. Select ISMF option 7.4, ACS Test application to display the ACS Test Case Define/Alter panel.

- 
2. Specify the security label value in the SECLABEL field. You also can specify other values that you want to test in the ACS routines.

- 
3. To run the test cases in a test library, specify the library name and select option 3 in the ACS Test Selection panel. On the Test Case Routines panel, indicate which routines you want to test.

---

**Tip:** If you are using NaviQuest, select option 11, selection 1.1 to specify the security label value in the Test Case Generation from Saved ISMF List Entry panel.

---

## Chapter 14. Activating Storage Management Subsystem configurations

You can activate an SMS configuration manually, or automatically at IPL. This topic 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 11.

#### 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 additional information, see [z/OS MVS Initialization and Tuning Guide](#).

**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 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, 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, the following command indicates that the ACDS specified in IGDSMS02 contains the SMS configuration to be activated at future IPLs:

```
SMS,IGDSSIIN,'ID=02,PROMPT=DISPLAY'
```

## Converting the SMS configuration from compatibility to 32-system mode

---

When converting to 32-system mode, you must convert all systems in the SMSplex at the same time. Use the procedure that follows.

**Note:** The procedure indicates that the ACDS is converted first, followed by the COMMDS. However, due to the nature of parallel tasks in SMS, these conversions may be reversed, so that the COMMDS is converted first.

1. Set a date and time for the conversion when the allocation activity on the system is low.
2. Create an SCDS, ACDS, and COMMDS by copying existing SMS control data sets. Ensure that you have the correct data set space allocations. Refer to [“Allocating control data sets” on page 11](#) for space requirements for converting to 32-system mode.

To provide for emergency fallback, do not convert your existing SCDS, ACDS, and COMMDS.

3. On each system in the SMSplex, create an IGDSMSxx member of SYS1.PARMLIB that points to the new ACDS and COMMDS and specifies SYSTEMS(32).

4. Immediately before conversion, issue the command SETSMS INTERVAL(999) on all systems sharing the ACDS and COMMDS. This prevents a system from rereading the COMMDS and attempting to activate the new configuration before you have an opportunity to change the mode of that system.
5. On one system in the SMSplex, issue the command SET SMS=xx to restart SMS or re-IPL the system using the new IGDSMSxx member of SYS1.PARMLIB. This restarts SMS in 32-system name mode using the new ACDS and COMMDS. The following sequence of messages will be received on this system that accesses the new control data sets:

```
IGD064I ACDS dsname SUPPORTS ONLY 8 SYSTEMS
IGD092I WARNING: YOU MUST BE PREPARED TO CONVERT ALL SYSTEMS IN
        THE SMSPLEX. FAILURE TO DO SO MAY CAUSE ERRORS ACCESSING
        THE CONFIGURATION DATA SET(S).
IGD067D REPLY 'CONVERT' TO ALLOW CONVERSION OF ACDS TO SUPPORT
        MORE THAN 8 SYSTEMS OR 'REJECT' TO FAIL THE REQUEST
```

Reply 'CONVERT' to complete the conversion of the ACDS.

```
IGD039I ACDS dsname HAS BEEN CONVERTED TO SUPPORT MORE THAN 8
        SYSTEMS.
IGD009I ACDS SWITCHED TO dsname
```

**Note:** At this point, your ACDS has been converted and you are about to convert the COMMDS.

```
IGD064I COMMDS dsname SUPPORTS ONLY 8 SYSTEMS
IGD092I WARNING: YOU MUST BE PREPARED TO CONVERT ALL SYSTEMS IN
        THE SMSPLEX. FAILURE TO DO SO MAY CAUSE ERRORS ACCESSING
        THE CONFIGURATION DATA SET(S).
IGD067D REPLY 'CONVERT' TO ALLOW CONVERSION OF COMMDS TO SUPPORT
        MORE THAN 8 SYSTEMS OR 'REJECT' TO FAIL THE REQUEST
```

Reply 'CONVERT' to complete the conversion of the COMMDS.

```
IGD039I COMMDS dsname HAS BEEN CONVERTED TO SUPPORT MORE THAN 8
        SYSTEMS.
IGD009I COMMDS SWITCHED TO dsname
```

6. On the remaining systems in the SMSplex, issue the command SET SMS=xx to restart SMS or re-IPL the system using the new IGDSMSxx member of SYS1.PARMLIB for that system. This restarts SMS in 32-system name mode using the new ACDS and COMMDS. The following sequence of messages will be received on this system that accesses the new control data sets:

```
IGD009I ACDS SWITCHED TO dsname
IGD009I COMMDS SWITCHED TO dsname
```

## Displaying storage management subsystem information

You can use the DISPLAY SMS MVS command to display information about the storage management subsystem, such as the following information:

- Active SMS configuration
- SMSVSAM status of sharing control data sets
- SMSVSAM server name
- Coupling facility cache and lock structures

For a description of the DISPLAY SMS command, see [z/OS MVS System Commands](#).

## Controlling DFSMStvs processing

While DFSMStvs is processing, you can monitor the performance of applications programs and maintain data integrity during backup-while-open (BWO) processing.

## Monitoring application programs that use DFSMStvs

For information about monitoring application programs that use DFSMStvs and tuning performance, see [z/OS DFSMStvs Planning and Operating Guide](#).

## Changing DFSMStvs status

You can use the VARY SMS MVS command to change the status for DFSMStvs in these ways:

- Change the state of a DFSMStvs instance or change the state of all DFSMStvs instances in the sysplex
- Change the state of a log stream to which DFSMStvs has access
- Change the state of a data set for VSAM record-level sharing (RLS) and DFSMStvs access
- Start or stop peer recovery processing for a DFSMStvs instance

**Restriction:** You cannot use the VARY SMS command to change the state of a DFSMStvs instance while it is initializing. Any attempt to do so is suspended until the initialization completes.

The possible states of a DFSMStvs instance follow:

### ENABLE

Enables DFSMStvs to begin accepting new units of recovery for processing.

### DISABLE

Prevents DFSMStvs from processing new work requests. DFSMStvs does not process new work requests from units of recovery that are currently in progress.

### QUIESCE

Prevents DFSMStvs from accepting any new units of recovery for processing. DFSMStvs completes the processing of any units of recovery in progress.

The possible states of a data set follow:

### ENABLE

Unquiesces a data set for VSAM RLS and DFSMStvs access.

### DISABLE

Quiesces a data set for VSAM RLS and DFSMStvs access.

For a description of the VARY SMS command, see [z/OS MVS System Commands](#).

For information about the effects of DFSMStvs, log stream, and data set states on DFSMStvs processing, see [z/OS DFSMStvs Planning and Operating Guide](#).

## Changing Storage Management Subsystem parameters

You can use the SET SMS or SETSMS MVS command to change SMS parameters. Use the SET SMS command to initialize SMS parameters from PARMLIB member IGDSMSxx and start SMS (or restart SMS if it was already active). Use the SETSMS command when SMS is active (running) to change a subset of SMS parameters by entering them from the console without changing IGDSMSxx.

For descriptions of SET SMS and SETSMS, see [z/OS MVS System Commands](#).

**Exception:** To reactivate a null configuration, you must perform an IPL.

**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” on page 182 lists only those parameters that can be changed using the SETSMS operator command. For additional information, see [z/OS MVS Initialization and Tuning Guide](#)

## Parameters of the SETSMS operator command

You can change SMS parameters by using the SETSMS operator command. For descriptions of the parameters, see [z/OS MVS System Commands](#) and [z/OS MVS Initialization and Tuning Guide](#).



## SETSMS command to alter the setting of BreakPointValue

You can use the SETSMS command to change the setting of BreakPointValue without having to re-IPL. This modified setting is in effect until the next IPL when it reverts to the value specified in the IGDSMSxx member of PARMLIB. To make the setting change permanent, you must alter the value in SYS1.PARMLIB.

The syntax of the operator command is:

```
SETSMS BreakPointValue(0-65520)
```

**Note:** You cannot use the SETSMS command to alter the value inside any individual Storage Group. You must use the ALTER command in the ISMF Storage Group Application.

## SETSMS command to alter the setting of USEEAV

You can use the SETSMS command to change the setting of USEEAV without having to re-IPL. This modified setting is in effect until the next IPL when it reverts to the value specified in the IGDSMSxx member of PARMLIB. To make the setting change permanent, you must alter the value in SYS1.PARMLIB.

The syntax of the operator command is:

```
SETSMS USEEAV(YES|NO)
```

The USEEAV(NO) setting indicates that no new primary space allocations or extended data sets to a new volume are allowed on an EAV.

## 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 recommended alternative for keeping the SMS configuration information.

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 15. Maintaining the Storage Management Subsystem

After activating SMS, you need to monitor and adjust it over time. This topic explains how to maintain SMS. Maintenance activities include listing, altering and ejecting optical and tape volumes. Maintenance activities also include listing, altering, copying, and deleting the following SMS components:

- Storage groups
- Management classes
- Storage classes
- Data classes
- Copy pools
- Aggregate groups
- Optical libraries
- Optical drives
- Tape libraries

**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 Chapter 24, “Using NaviQuest,” on page 323 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.

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 19 on page 186](#).

```

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    2004/12/01 09:23:29        15
  SYSTEM02    -----
  SYSTEM03    -----
  SYSTEM04    -----
  SYSTEM05    -----
  SYSTEM06    -----
  SYSTEM07    -----
  SYSTEM08    -----

```

Figure 19. 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 [z/OS MVS Initialization and Tuning Guide](#)

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 20 on page 186](#).

```

14.57.24          IGD002I 14:57:24 DISPLAY SMS 879

TRACE      = {ON|OFF}   SIZE =nnnK      TYPE {ERROR|ALL}
JOBNAME = {j|j|*}      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 20. 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](#) for additional information.

## Displaying storage group status using the DISPLAY SMS command

You can use the STORGRP parameter to display the status of storage groups including tape and object storage groups. [Figure 21 on page 187](#) 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 21. 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 22 on page 187](#).

```
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 22. 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 23 on page 188](#).

```
D SMS,STORGRP(SG1),LISTVOL

15.12.25          d sms,storgrp(slp01),listvol
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 MVS SYSTEM= 1 2 3 4 5 6 7 8 STORGRP NAME
123456   1256 ONRW      + . . . . . SG1
123457   1257 ONRO      + . . . . . SG1
123458   1258 PORW      + . . . . . SG1
123459   1259 PORO      + . . . . . SG1
123460   1260 OFRW      + . . . . . SG1
123461   1261 OFRO      + . . . . . 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 23. Displaying Storage Group Volume Status Information

MVS status column values are as follows:

### **BX**

The device is boxed.

### **NR**

The device is not ready.

### **OF**

The device is offline.

### **ON**

The device is online.

### **PO**

The device is pending offline.

Access-mode values are as follows:

### **RO**

The device permits read-only access.

### **RW**

The device permits read-write access.

## 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 24. 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.

Perform the following steps to display a storage group's status.

1. Display the Storage Group Application Selection panel (select option 6 from the ISMF Primary Option Menu for storage administrators).

- 
2. Supply values to create a list of storage groups. For example, you could use the following to create a list of storage groups with names that begin with SGNAME:

- CDS Name: 'ACTIVE'
- Storage Group Name: SGNAME\*
- Option: 1, List. This displays the Storage Group List panel.

- 
3. Type LISTSYS in the Line Operator column next to the desired pool type storage group on the Storage Group List panel. This shows the SMS Storage Group Status Display.

**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.

**Note:** The information provided by LISTSYS differs from that provided by LISTVOL, because the information for the two commands comes from different sources.

The Total Space for Storage Group value is the total capacity of all the volumes belonging to storage group SGNAME01 where SMS volume status is ENABLE, MVS volume status is ONLINE, and the volume is not read-only.

**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. The fields contain values only if SMS volume status is ENABLE, MVS volume status is ONLINE, and the volume is not read-only.

**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

Perform the following steps to display a volume's status.

1. Display the Storage Group Application Selection panel (select option 6 from the ISMF Primary Option Menu for storage administrators).

- 
2. Select option 4, Volume. This displays the Storage Group Selection panel.

- 
3. Supply values on the Storage Group Selection panel:

- a. The single volume or range of volumes you wish to display.



- b. Option 1, Display, to display the SMS Volume Status Display panel.
- 

## Values on the SMS volume status display panel

The SMS Volume Status Display panel shows status information.

### Physical Volume Status

This 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.

### MB-free

This is the total number of free megabytes belonging to the volume.

### % Free

This 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.

### SMS Vol Status

This 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.

### MVS Vol Status

This 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.

## READONLY

The device is online and will only accept read requests.

## 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 25 on page 192](#). In this example, the volume serial number is 123456.

```
15.38.00          IGD002I 15:38:00 DISPLAY SMS 918
VOLUME UNIT MVS SYSTEM= 1 2 3 4 5 6 7 8 STORGRP NAME
123456 1256 ONRW      + . . . . . SG1
*****
. 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 25. Displaying Status Information for Individual DASD Volumes

See “Displaying the status of volumes in the storage group” on [page 188](#) for a list of MVS status and access mode values that might appear.

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.

Use the following 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 26 on page 193 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      MD VOLSER VOLSTAT  STORGRP SGSTAT
 430 ,3380      ,OR,XP0101,ENABLED,  SXP01  ,QUIESCED

*****LEGEND*****
A = ALLOCATED          F = OFFLINE
M = MOUNT PENDING     N = NOT ALLOCATED
O = ONLINE            P = PENDING OFFLINE
R = READ-ONLY

Note: Tape device type is not supported by DEVSERV SMS
```

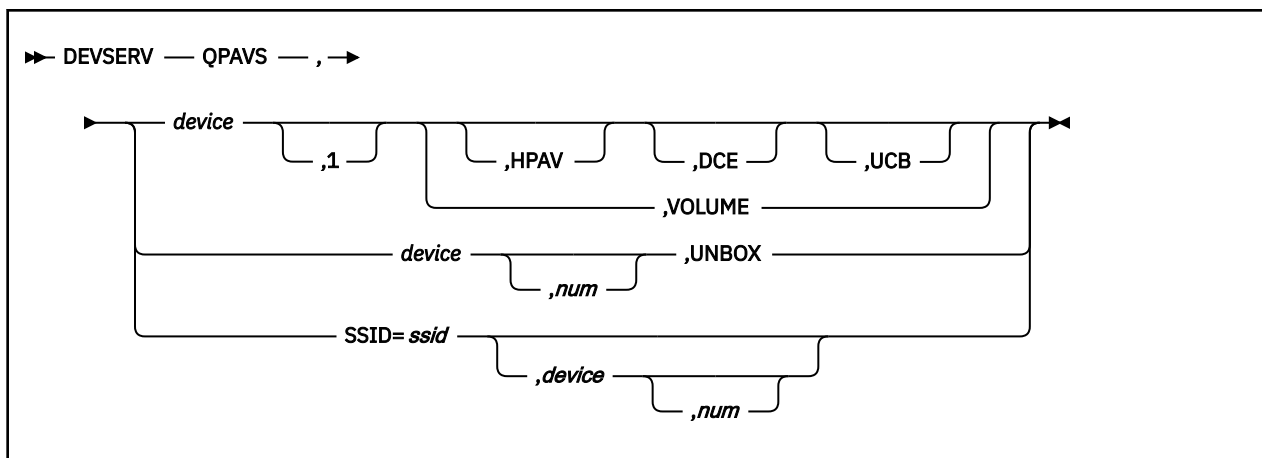
Figure 26. DEVSERV S,430 Output

## **DEVSERV QPAVS**

DEVSERV QPAVS supports the parallel access volume (PAV) capability of the IBM Enterprise Storage Server (ESS). You can use the DEVSERV QPAVS command to perform the following tasks:

- 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

Use the following syntax for the DEVSERV QPAVS command:



Where:

### QPAVS

Is a required positional keyword. May be abbreviated as QPAVS or QP.

### device

Specifies the device or devices to be displayed. *device* can be specified in either of the following formats:

- *sccuu*, where *s* is either 0 or 1 to indicate the desired subchannel and *ccuu* specifies a 4 hex digit device number (3-digit device numbers must be padded with a leading zero), or
- *ccuu*, which specifies just the device number. In this case, the active subchannel set is used to return information about the device.

### num

Specifies the number of devices to be displayed. *num* can be a decimal number from 1 to 256. 1 is the default.

### DCE

Displays the device class extension block (DCE) of the BASE UCB. If DCE is specified, only one device may be displayed (that is *num* must be 1).

### UCB

Displays the unit control block (UCB) information associated with the device. If UCB is specified, only one device may be displayed (that is *num* must be 1).

### UNBOX

Causes QPAVS to unbox the unbound alias device if it is in a BOX state. *num* can be a decimal number from 1 to 256.

### VOLUME

Displays the parallel access volume (PAV) relationship information for the logical volume, including the PAV base device number and all PAV alias device numbers bound to that base. May be abbreviated as VOL.

### HPAV

Displays the number of alias pool devices or the alias pool device numbers in the same logical control unit as a base device number (*sccuu*) that is the target of the QPAV command. If the *sccuu* number is issued to a HyperPAV alias device, only that device will be in the output.

### SSID=ssid

Specifies the subsystem identification number (SSID) of the subsystem whose information DEVSERV is to display.

**Note:** If a specific set of devices is specified after SSID with the *device* and *num* parameters, then IO will be issued only to those devices.

Figure 27 on page 195 shows the display content when you issue the DEVSERV QPAVS command with parameters other than DCE and UCB. If you specify the DCE or UCB parameter (or both), additional DCE

or UCB information (or both) is displayed for the device. This additional information is formatted the same way as the output resulting from the QDASD parameter.

```

-----
---          DS QP  Display Content          ---
-----
IEE459I  (time)  DEVSERV QPAVS
Host Configuration                      Subsystem Configuration
-----
UNIT   UA   TYPE      STATUS      SSID   ADDR   UA
NUM.  UA   TYPE      STATUS      SSID   ADDR   UA
-----
sdddd aa   BASE      INV-ALIAS  ssss   uu   BASE
                ALIAS-bbbb NOT-BASE  ssss   uu   ALIAS-bb
                NON-PAV  NOT-ALIAS ssss   uu   NC
sdddd bb   ALIAS      UNBOUND  ssss   uu   ALIAS-aa
sdddd bb   ALIAS-bbbb  ssss   uu   ALIAS-aa
sdddd aa   BASE-H      ssss   uu   BASE
sdddd bb   ALIAS-H      ssss   uu   ALIAS-H

UNLISTED DEVICES AND REASON CODES X
sdddd(rc) sdddd(rc) sdddd(rc) ...

```

Figure 27. 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
- (0B)**  
DEVICE IS A SECONDARY OF A PPRC PAIR
- (0C)**  
SUBCHANNEL SET VALUE SPECIFIED IS NOT VALID
- (0D)**  
UCB NOT FOUND IN SPECIFIED SUBCHANNEL SET
- (0E)**  
DEVICE IS A HYPERPAV ALIAS
- (0F)**  
DEVICE IS NOT A HYPERPAV BASE OR ALIAS

The following are examples of DEVSERV QPAVS output.

Figure 28 on page 196 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  TYPE      STATUS  SSID  UNIT  UA
NUM.  ADDR. ADDR. TYPE
-----
0D123 23  NON-PAV
0D124 24  NON-PAV
0D125 25  NON-PAV
****  3  DEVICE(S) MET THE SELECTION CRITERIA

```

Figure 28. DS QP,D123,3 Output

Figure 29 on page 196 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  TYPE      STATUS  SSID  UNIT  UA
NUM.  ADDR. ADDR. TYPE
-----
0D222 22  BASE
0D2FE FE  ALIAS-D222
0D2FF FF  ALIAS-D222
****  3  DEVICE(S) MET THE SELECTION CRITERIA

```

Figure 29. DS QP,D2FF,VOLUME Output

Figure 30 on page 196 shows the PAV status for the base device at the address D222 and its alias volumes without the subchannel set support installed.

```

IEE459I 08.20.32 DEVSERV QPAVS 591
      Host      Subsystem
      Configuration      Configuration
-----
UNIT  UNIT  UA  TYPE      STATUS  SSID  UNIT  UA
NUM.  ADDR. ADDR. TYPE
-----
0D222 22  BASE
0D2FE FE  ALIAS-D222
0D2FF FF  ALIAS-D222
****  3  DEVICE(S) MET THE SELECTION CRITERIA

```

Figure 30. DS QP,D222,VOLUME Output

Figure 31 on page 196 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 SUCCESSFULLY UNBOXED 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 31. 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 32 on page 197 describes the status of an unbound alias and the display of its associated UCB control blocks.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
      Host
      Configuration
      -----
      UNIT
      NUM. UA  TYPE          STATUS
      -----
      UCB AT V01ED2AA8
      0088FF04D5EF0000 00000000008E4C3C2 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:
      0D5EF(0A)
      **** (0A) - DEVICE IS AN UNBOUND PAV-ALIAS
```

Figure 32. DS QP,D5EF,UCB Output

Figure 33 on page 197 shows the PAV status of the devices that have the same SSID value.

```
IEE459I 08.20.32 DEVSERV QPAVS 591
      Host
      Configuration
      -----
      UNIT
      NUM. UA  TYPE          STATUS
      -----
      0D400 00  NON-PAV
      ....
      0D422 22  BASE
      ....
      0D4FF FF  ALIAS-D422
      **** 256 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 33. DS QP,SSID=1401 Output

Figure 34 on page 197 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
      Configuration
      -----
      UNIT
      NUM. UA  TYPE          STATUS
      -----
      0D345 45  BASE          NOT-BASE
      **** 1 DEVICE(S) MET THE SELECTION CRITERIA
```

Figure 34. DS QP,D345 Output

Figure 35 on page 198 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 Configuration          Subsystem Configuration
-----
UNIT  UNIT  UA  TYPE  STATUS  SSID  ADDR.  UA  TYPE
NUM.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.
-----
0D621 21  ALIAS-D600  NOT-ALIAS  0106  21  BASE
**** 1 DEVICE(S) MET THE SELECTION CRITERIA

```

Figure 35. DS QP,D621 Output

Figure 36 on page 198 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 Configuration          Subsystem Configuration
-----
UNIT  UNIT  UA  TYPE  STATUS  SSID  ADDR.  UA  TYPE
NUM.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.
-----
0D6F4 F4  ALIAS-D600  INV-ALIAS  0106  F4  ALIAS-06
**** 1 DEVICE(S) MET THE SELECTION CRITERIA

```

Figure 36. DS QP,D6F4 Output

Figure 37 on page 198 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 Configuration          Subsystem Configuration
-----
UNIT  UNIT  UA  TYPE  STATUS  SSID  ADDR.  UA  TYPE
NUM.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.
-----
00F60 F0  NON-PAV  NOT-NPAV  0101  F0  ALIAS-03

```

Figure 37. DS QP,F60 Output

Figure 38 on page 198 shows the display content when you issue the DEVSERV QPAVS command with the HPAV parameter, with the target of the command being a HyperPAV alias device number.

```

09.07.08 SYSTEM1          ds qpavs,e27f,hpav
09.07.08 SYSTEM1          IEE459I 09.07.08 DEVSERV QPAVS 472
Host Configuration          Subsystem Configuration
-----
UNIT  UNIT  UA  TYPE  STATUS  SSID  ADDR.  UA  TYPE
NUM.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.  ADDR.
-----
0E27E  ALIAS-H
0E27F  ALIAS-H
**** 2 DEVICE(S) IN HYPERPAV ALIAS POOL

```

Figure 38. DS QP,E27F,HPAV Output

Figure 39 on page 199 shows the display content when you issue the DEVSERV QPAVS command with the HPAV parameter, with the target of the command being a HYPERPAV base device number.



```

09.07.08 SYSTEM1          ds qpavs,e200,hpav
09.07.08 SYSTEM1          IEE459I 09.07.08 DEVSERV QPAVS 472
      HOST                SUBSYSTEM
      CONFIGURATION        CONFIGURATION
-----
UNIT  UNIT  UA  TYPE      STATUS      SSID  ADDR.  UA
NUM.  NUM.  UA  TYPE      STATUS      SSID  ADDR.  TYPE
-----
0E200 00  BASE-H          -----  3205  00    BASE
0E27E          ALIAS-H
0E27F          ALIAS-H
****          3 DEVICE(S) IN HYPERPAV ALIAS POOL

```

Figure 39. DS QP,E200,HPAV Output

Figure 40 on page 199 shows the display content when you issue the DEVSERV QPAVS command with the VOLUME parameter, with the target of the command being a HYPERPAV alias device number.

```

09.07.08 SYSTEM1          ds qpavs,e27f,volume
09.07.08 SYSTEM1          IEE459I 09.07.08 DEVSERV QPAVS 472
      HOST                SUBSYSTEM
      CONFIGURATION        CONFIGURATION
-----
UNIT  UNIT  UA  TYPE      STATUS      SSID  ADDR.  UA
NUM.  NUM.  UA  TYPE      STATUS      SSID  ADDR.  TYPE
-----
**** UNLISTED DEVICE(S) AND REASON CODES :
0E27F(0E)
**** (0E) - DEVICE IS A HYPERPAV ALIAS

```

Figure 40. DS QP,E27F,VOLUME Output

## DEVSERV PATHS

You can use the DEVSERV PATHS command to display the number of configured cylinders for a device or range of devices under CYL header. If a device is part of a DUAL COPY pair (devices behind a 3990-3 or older), then the ALT dddd is displayed under the CYL header. The first character of the UNIT number represents the subchannel set number.

Figure 41 on page 199 shows the output of the DEVSERV P,F4A command.

```

18.09.58 SYSTEM1          ds p,0f4a,4
18.09.58 SYSTEM1          IEE459I 18.09.58 DEVSERV PATHS 600
      UNIT DTYPE M CNT VOLSER  CHPID=PATH STATUS
      RTYPE SSID CFW TC  DFW PIN DC-STATE CCA DDC      CYL CU-TYPE
00F4A,33909 ,0,000,EV9LIA,00=+ 04=+
2105 201F Y YY. YY. N SIMPLEX 3A      10017 2105
***** SYMBOL DEFINITIONS *****
0 = ONLINE          + = PATH AVAILABLE

```

Figure 41. DEVSERV P,F4A Output

Path Attributes Definitions:

### NS (Not Specified)

Accessibility not specified

### NP (Non Preferred)

The control unit image that was selected by this command is accessible over the interface associated with this Interface ID, but the interface is a non-preferred path for this control unit image.

### PF (Preferred)

The control unit image that was selected by this command is accessible over the interface associated with this Interface ID, and the interface is a preferred path for this control unit image.

## DEVSERV QDASD

You can use the DEVSERV QDASD command to display the number of configured cylinders for a device or range of devices, including device model 3390 model A. The first character of the UNIT number represents the subchannel set number.

Figure 42 on page 200 shows the output of the DS QD,F41,RDC,DCE command.

```

09.36.34 SYSTEM1          IEE459I 09.36.34 DEVSERV QDASD 625
UNIT VOLSER SCUTYPE DEVTYPE      CYL  SSID SCU-SERIAL DEV-SERIAL EFC
00F41 IN7996 2107921 2107900      87927 2606 0113-03261 0113-03261 *OK
READ DEVICE CHARACTERISTIC
2107E833900E5E8C FFF72032FFFE000F E000E5A205940222 1309067400000000
0000000000000000 3E3E1F02DFEE0001 0677080F007F4A00 003C000000015777
DCE AT V00E8A578
4878807100000000 0000000002051820 D800FFFEFFFE3E3E 1FF70A00002A0000
00FD7B9C9400F0FE 161F3C1E000A8C00 1500000000015777 0001577600000000
0000000000000000
****      1 DEVICE(S) MET THE SELECTION CRITERIA
****      0 DEVICE(S) FAILED EXTENDED FUNCTION CHECKING

```

Figure 42. DS QD,F41,RDC,DCE Output

When you add the ATTR parameter, DEVSERV QDASD displays device attributes, including if the device is a solid state drive, for example:

UNIT	VOLSER	SCUTYPE	DEVTYPE	CYL	SSID	SCU-SERIAL	DEV-SERIAL	EFC
00300	TK9085	2107921	2107900	65520	2401	0175-02411	0175-02411	*OK
ATTRIBUTE/FEATURE				YES/NO				
SOLID STATE DRIVE				Y				
ATTRIBUTE/FEATURE				YES/NO				
ENCRIPTION				N				

## 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. Chapter 20, “Quick reference to ISMF commands and line operators,” on page 299 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
- 62 x 160

The examples shown in this document 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, Copy Pool, 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” on page 201 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 perform the following steps.

1. Select option 4, Data Class, from the ISMF Primary Option Menu for Storage Administrators. This displays the Display the Data Class Application Selection panel.

- 
2. Supply values on the Data Class Application Selection panel:

**Option**

Select option 1, List.

**Respecify View Criteria**

Select the viewing order by typing / for this field.

**Respecify Sort Criteria**

Select the sorting order by typing / for this field.

This displays the Data Class View Entry panel.

- 
3. Supply values on the Data Class View Entry panel. This panel shows the columns that you can select.

Use DOWN to see the remaining pages for this panel.

At any time, you can choose option 2, Save, to save your specifications for later use.

To select attributes, type the corresponding "tags," separated by blanks, in the Specify Tags in Sequence Desired field. The tags are the numbers enclosed in parentheses for each column, for example, (26) Data Set Name Type. For example, to select Data Set Name Type, Logical Record Length, and Record Organization, you would type:

```
Specify tags in Sequence Desired:
===> 26 5 3
```

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.

Press Enter to see the Data Class Sort Entry panel.

- 
4. Specify values for sorting on the Data Class Sort Entry panel. Specify a field by its numerical tag. 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.

Press Enter to see a list of data classes that uses your view and sort criteria.

On the list panel, 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.

You can scroll up, down, left, and right. However, ISMF always displays the LINE OPERATOR and DATA CLAS 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 components

You can change the definitions of SMS components, libraries, volumes and drives 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 components. This method is more efficient if you want to make multiple changes in one SCDS.

You can also use the MVS SETSMS command to alter some components.

This topic outlines how to alter SMS components. For information on how to alter libraries and drives, see the following:

- Altering optical libraries: [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support](#)
- Altering tape libraries and drives: [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries](#)

This topic outlines how to alter each SMS component using the component's Application Selection panel.

**Note:** You must be in storage administrator mode to alter SMS classes. If you are in user mode, you are not authorized to alter SMS classes and libraries. [“Storage administrator authorization”](#) on page 233 describes storage administrator mode.

SMS Component	Associated procedure (see . . . )
Storage Group	<ul style="list-style-type: none"><li>• <a href="#">“Altering storage groups”</a> on page 203</li><li>• <a href="#">“Redefining volumes among storage groups”</a> on page 203</li></ul>
Management Class	<a href="#">“Altering management classes”</a> on page 204
Storage Class	<a href="#">“Altering storage classes”</a> on page 205
Data Class	<a href="#">“Altering data classes”</a> on page 205
Copy Pool	<a href="#">“Altering copy pools”</a> on page 206
Aggregate Group	<a href="#">“Altering aggregate groups”</a> on page 207

## 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, and specify the name of the SCDS containing the storage group that 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 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 command.

## Redefining volumes among storage groups

Perform the following steps if you defined DASD volumes to belong to one storage group but you now want them to belong to a different storage group.

1. Display the Storage Group Application Selection panel (select option 6 from the ISMF Primary Option Menu for storage administrators).

- 
2. Supply values on the Storage Group Application Selection panel:

**CDS Name**

Is the name of the SCDS.

**Storage Group Name**

Is the name of the storage group.

**Option**

Select option 4, Volume.

This displays the Storage Group Volume Selection panel.

- 
3. Supply values on the Storage Group Volume Selection panel:

**Specify a Single Volume (in Prefix), or Range of Volumes**

Type the serial numbers of the volumes you want to redefine.

**Option**

Select option 4, Delete.

Press Enter to delete the volumes from their current storage group.

- 
4. Return to the Storage Group Application Selection panel.

- 
5. Supply values on the Storage Group Application Selection panel:

**Storage Group Name**

Is the name of the new storage group to contain the volumes.

**Option**

Select option 4, Volume. This returns you to the Storage Group Volume Selection panel.

- 
6. Supply values on the Storage Group Volume Selection panel:

- Specify the volume serial numbers.
- Select option 2 to define them to the storage group.

Press Enter to process the values.

---

7. Activate the updated configuration.

**Result:** The changes take effect when you activate this updated configuration.

You are responsible for maintaining consistent physical connections. You must also be aware of multivolume data sets. In general, all of a multivolume data set must be in a single storage group (it is possible, when the primary storage group is stressed, for a multivolume data set to extend to more than one 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 MVS SETSMS command to change the SMS status of volumes. Any SMS status that is changed by a SETSMS command is effective only until a new configuration is activated. The MVS status is not affected by either the SETSMS command or the activation of a new configuration. Only the SMS status is affected.

## Altering management classes

As the needs of your installation change, you might need to modify management classes.

Perform the following steps to modify management classes.

1. Display the Management Class Application Selection panel (select option 3 from the ISMF Primary Option Menu for storage administrators).

- 
2. Supply values on the Management Class Application Selection panel:

**CDS Name**

Is the name of the SCDS that contains the management class you want to change.

**Management Class Name**

Is the name of the management class.

**Option**

Select option 4, Alter. This displays the Management Class Alter panel. This panel contains the same attributes as the Management Class Define panel. The fields are primed with the information that you entered when you defined the management class.

- 
3. Type over the old information with your new information on the Management Class Alter panel. Use the DOWN command to see the next pages. See [“Values on the management class alter panel” on page 204](#) for more information.

- 
4. Use the END command to save the changes and return to the Management Class Application Selection panel.

**Result:** The changes take effect when you activate this updated configuration.

## Values on the management class alter panel

The following list describes the fields that are specified on the Management Class Alter panel:

**# Versions**

Specifies the number of versions of data sets that are assigned this management class, that are to be maintained.

**Retain Only/Extra Version(s)**

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.

**Restriction:** This field cannot be left blank if the Unit field is specified.

### Copy Serialization

Specifies whether aggregate backup should continue even if the data set is allocated to a job.

### Abackup Copy Technique

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 that are 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.

## Altering storage classes

As the needs of your complex change and as new hardware becomes available, you might need to change the storage classes.

Perform the following steps to modify a storage class.

1. Display the Storage Class Application Selection panel (select option 5 from the ISMF Primary Option Menu for storage administrators).

- 
2. Supply values on the Storage Class Application Selection panel:

#### CDS Name

Is the name of the SCDS that contains the storage class you want to change.

#### Storage Class Name

Is the name of the storage class.

#### Option

Select option 4, Alter. This displays the Storage Class Alter panel. This panel contains the same attributes as the Storage Class Define panel. The fields are primed with the information that you entered when you defined the storage class.

- 
3. Type over the old information with your new information on the Storage Class Alter panel. For more information, see [“Defining storage class attributes” on page 84](#).

Use the DOWN command to see the next page.

- 
4. Use the END command to save the changes and return to the Storage Class Application Selection panel.

**Result:** 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

Perform the following steps to modify a data class.

1. Display the Data Class Application Selection panel (select option 4 from the ISMF Primary Option Menu for storage administrators).
-

2. Supply values on the Data Class Application Selection panel:

**CDS Name**

Is the name of the SCDS that contains the data class you want to change.

**Data Class Name**

Is the name of the data class.

**Option**

Select option 4, Alter. This displays the Data Class Alter panel. This panel contains the same attributes as the Data Class Define panel. The fields are primed with the information that you entered when you defined the data class.

---

3. Type over the old information with your new information on the Data Class Alter panel. For more information, see [“Defining data class attributes” on page 118](#).

Use the DOWN command to see the next page.

---

4. Use the END command to save the changes and return to the Data Class Application Selection panel.

**Result:** The changes take effect when you activate this updated configuration. The changes you make apply only to new data sets. The following features of the existing data sets are affected: additional volume amount, extent constraint relief, and dynamic volume count.

In addition, page 1 of this panel allows 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).

For descriptions of the attributes contained in the Data Class Alter and the Data Class Define panels, see [“Defining VSAM attributes and specifying media types for data class” on page 128](#).

## Altering copy pools

You can use the Copy Pool Alter panel to add or remove storage groups from the copy pool. You can also use this panel to alter the number of backup versions.

Perform the following steps to modify a copy pool.

1. Display the Copy Pool Application Selection panel (select option P from the ISMF Primary Option Menu for storage administrators).
- 

2. Supply values on the Copy Pool Application Selection panel:

**CDS Name**

Is the name of the SCDS that contains the copy pool.

**Copy Pool Name**

Is the name of the copy pool.

**Option**

Select option 4, Alter. This displays the Copy Pool Alter panel. The Copy Pool Alter panel contains the same attributes as the Copy Pool Define panel. The fields are primed with the information you entered when you defined the copy pool.

---

3. Type over the old information with your new information on the Copy Pool Alter panel. For more information, see [“Defining a copy pool” on page 144](#).

Use the DOWN command to see the next page.

---



4. Use the END command to save the changes and return to the Data Class Application Selection panel.

## 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 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.

Perform the following steps to modify an aggregate group.

1. Display the Aggregate Group Application Selection panel (select option 9 from the ISMF Primary Option Menu for storage administrators).

- 
2. Supply values on the Aggregate Group Application Selection panel:

**CDS Name**

Is the name of the SCDS containing the aggregate group you want to alter.

**Aggregate Group Name**

Is the name of the aggregate group.

**Option**

Select option 4, Alter. Press Enter to see the Aggregate Group Alter panel. It contains the same attributes as the Aggregate Group Define panels discussed in [“Defining aggregate group attributes” on page 139](#). The fields are primed with the information you entered when defining the aggregate group.

- 
3. Type over the old information with your new information on the Aggregate Group Alter panel. For more information, see [“Defining aggregate group attributes” on page 139](#).

Use the DOWN command to see the next page.

- 
4. 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.

## Editing Aggregate group attributes

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 range from 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 can not be a digit) or a blank.

The second page of the Aggregate Group Alter panel contains the selection 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.

You can specify the following required attributes on the second page of the Aggregate Group Alter panel:

### To Edit a Data Set, Specify Number

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 9](#) on page 140. See [z/OS ISPF User's Guide Vol I](#) for more information on PDF Edit commands.

### Selection Data Sets

The name of the data set containing the 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. If you want to enter a fully qualified data set name, enclose the name in single quotation marks. If you do not enclose the name in single quotation marks, the TSO/E prefix is added to the name as the first high level qualifier.

### Member Name

The name of the data set member containing the 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/E 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

The name of the data set containing instruction, commands, and such, 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/E data set name. The data set name, including the TSO/E prefix, can be no more than 44 characters long. This is an optional field and has no default.

## Altering the SCDS on different DFSMS releases

The SCDS can be updated on other 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 to the most current level of the system. Back up the SCDS and ACDS as you would for any other production data set for the purpose of recovery. See [z/OS Upgrade Workflow](#) for more information about the PTFs required and steps to take before you can update the SCDS.

## Copying SMS components

---

To simplify the creation of new SMS classes, storage groups, aggregate groups, and SCDSs, you can copy existing ones and modify them.

When you copy pool storage groups, you can optionally copy the volume list associated with the storage group. For other types, the volume list is not copied.

**Before you begin:** Generate a list from an SCDS by selecting the LIST option on an application panel.

Perform the following steps to copy an SMS component.

1. In the LINE OPERATOR column of any List panel, type the COPY line operator and press Enter. You see the Copy Entry panel.

- 
2. Supply values for the fields on the Copy Entry panel:

#### **from Data Set Name**

Identifies the source you are copying. It is primed with the value you specified on the Application Selection panel.

#### **from Construct Name**

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.

#### **from Construct Type**

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 copy will have the same *type* as the original.

**to Data Set Name**

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.

**to Construct Name**

Identifies the name of the SMS class or storage group copy. It is primed with blanks.

**Replace like-named Construct**

If the SMS class, storage group, or aggregate group that you identify in the *to* fields already exists, you use this field to specify whether you want to replace it. If you select this option, the replacement occurs. By default the option is not selected, and the replacement does not occur.

**Perform Alter**

Indicates if you want to change some of the attributes of the copy you are creating. If you select this option, 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. By default, the option is not selected. You remain on the Copy Entry panel, where you can perform another copy or return to the original List panel.

**Copy Storage Group Volumes (Pool SG Only)**

Indicates if you want the volumes that are defined in the storage group to be copied. This option is valid only for pool storage groups. If you select this option, the volumes for the pool storage group are copied. They are defined to the new storage group as new volumes; no residual data is copied. By default, the option is not selected.

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 performing the following steps:

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”](#) on page 209 for information on deleting DASD volumes.

-----

4. Vary the DASD volume offline. See [z/OS DFSMSHsm Storage Administration](#) 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](#) 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. 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. Because 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. 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.

## Deleting DASD volume residual data

---

Each member in a SMS SYSPLEX maintains its own DASD Volume SMS and MVS status in the configuration. If you remove or delete a SMS member from the SMS SYSPLEX, its DASD volume's SMS and MVS data is not deleted from the shared ACDS or COMMDS. To clear this residual data from the configuration, you must allocate a new ACDS and COMMDS and then replace the current ones by activating the updated SCDS, SETSMS SCDS(scds\_name).

---

## Chapter 16. Recovering Storage Management Subsystem information

This topic 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 ISMF.

#### Restrictions and Recommendations:

1. 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.
2. 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.
3. 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.

If the SCDS and its backups are lost, you can save the ACDS as an SCDS using the SETSMS SAVESCDS command. For more information about how to use this command, see [“Parameters of the SETSMS operator command”](#) on page 182.

### Recovering an ACDS

#### About this task

There are situations where the ACDS may need to be recovered. 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.

If messages IGD041I and IGD040D with reason code 6069, indicating a DIV I/O error, appear on the operator console, you have the option of retrying the failing operation or using the spare ACDS.

When using the spare ACDS after messages IGD041I AND IGD040D with reason code 6069 are received, perform the following steps.

## Procedure

1. Reply C on the operator console.
2. Define a new linear VSAM data set with SHROPTNS(3,3), REUSE and ensure the new ACDS can be accessed by all systems in the SMS complex (that is, the volume is accessible from all systems).
3. 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(spareacdsdsn)
```

Do not use the SETSMS SAVEACDS command if the ACDS is corrupted.

4. Tell the system to use the spare ACDS by issuing the following command:

```
SETSMS ACDS(spareacdsdsn)
```

You need to issue the SETSMS ACDS(*spareacdsdsn*) command on only 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.

**Note:** In addition to the method previously described, you can also use the SETSMS COPYSCDS command to create an ACDS from any valid SCDS. This command may be especially useful in cases where the created ACDS is to be used on another system, such as in disaster recovery situations. Refer to [z/OS MVS System Commands](#) for more information on the SETSMS COPYSCDS command.

## Recovering an ACDS when messages IGD041I and IGD040D with reason code 6069 are not received

If the ACDS is corrupted or gets any permanent errors, but messages IGD041I and IGD040D with reason code 6069 are not received, you should recover the ACDS from a good backup copy of the SCDS.

## Procedure

1. Define a new linear VSAM data set with SHROPTNS(3,3), REUSE and ensure the new ACDS can be accessed by all systems in the SMS complex (that is, the volume is accessible from all systems).
2. Issue the SETSMS ACDS command to switch the sysplex over to the new ACDS.

```
SETSMS ACDS(newacdsdsn),SCDS(scdsdsn)
```

3. Ensure PARMLIB member IGDSMSxx is updated to reflect the new ACDS so that it will be used upon the IPL of any system in the sysplex.

## 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.commids)
```

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.commids*. You then need to issue the following command on each of the remaining systems in the SMS complex:

```
SETSMS COMMDS(spare.commids)
```



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 (such as, 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 17. 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 topic 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. If ACSDEFAULTS(YES), the data set key label is also covered by the RACF DATASET profile. Revoked USERID should not be used as a resource owner, or it causes RACF to fail. See also [“FIELD resource class”](#) on page 232.

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.

---

### 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
- Drive Configuration
- Data Collection
- Copy Pool
- List
- Mountable Tape Volume
- Tape Library

### 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/E 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 DFSMSdftp/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 15 on page 219 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 15. Module Names for ISMF Applications*

<b>Application</b>	<b>Module Name</b>
<b>Profile</b>	DGTFFPF00
<b>Data set</b>	DGTFDSD00
<b>DASD Volume</b>	DGTFFVA00
<b>Mountable Optical Volume</b>	DGTFFOVCD
<b>Mountable Tape Volume</b>	DGTFTVCD
<b>Management class</b>	DGTFFMCCD
<b>Data class</b>	DGTFFDCCD
<b>Storage class</b>	DGTFFSCCD
<b>Storage group</b>	DGTFFSGDR
<b>Automatic class selection</b>	DGTFFFLAD
<b>Control data set</b>	DGTFFSACD
<b>Aggregate Group</b>	DGTFFAGCD
<b>Optical Library Configuration</b>	DGTFFLCCD
<b>Optical Drive Configuration</b>	DGTFFRCCD
<b>Tape Library Configuration</b>	DGTFFLMCD
<b>Data Collection</b>	DGTFFADAD
<b>Report Generation</b>	DGTTHMD30
<b>List</b>	DGTFFJLCD
<b>Copy Pool</b>	DGTFFCPCD

## Locating module names for ISMF functions

Restricting access to the DGTFFPF05 module prevents end users from gaining access to the Primary Option Menu for Storage Administrators. [Table 16 on page 219](#) lists this and the other load module names you can use to limit access to certain ISMF functions.

*Table 16. Module Names for ISMF Functions*

<b>Function</b>	<b>Module Name</b>
User mode	DGTFFPF05
Logging and abend control	DGTFFPF02
ISMF job statement information	DGTFFPF03
DFSMSdss execute statement information	DGTFFPF04
ICKDSF execute statement information	DGTFFPF20
Data set print execute statement information	DGTFFPF21
IDCAMS execute statement information	DGTFFPF22

Table 16. Module Names for ISMF Functions (continued)

Function	Module Name
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	DGTFSGLD
Storage Group VOLUME	DGTFSGVR
Aggregate Group DEFINE	DGTFAGDA
Aggregate Group ALTER	DGTFAGAA
Aggregate Group DISPLAY	DGTFAGDI
Aggregate Group LIST	DGTFAGLD
Optical Library Configuration DEFINE	DGTFLCDE
Optical Library Configuration ALTER	DGTFLCAL
Optical Library Configuration DISPLAY	DGTFLCDI
Optical Library Configuration LIST	DGTFCLCD
Optical Drive Configuration DEFINE	DGTFRCDE
Optical Drive Configuration ALTER	DGTFRCAL
Optical Drive Configuration DISPLAY	DGTFRCDI
Optical Drive Configuration LIST	DGTFRCCLD
Tape Library Configuration DEFINE	DGTFMLDE
Tape Library Configuration ALTER	DGTFLMAL
Tape Library Configuration DISPLAY	DGTFMLDI

Table 16. Module Names for ISMF Functions (continued)

Function	Module Name
Tape Library Configuration LIST	DGTFMLMD
Copy Pool Define	DGTFCPDA
Copy Pool Alter	DGTFCPAA
Copy Pool Display	DGTFCPDI
Copy Pool List	DGTFCPDL

## Locating module names for line operators and commands

Table 17 on page 221, Table 18 on page 222, Table 19 on page 223 and Table 20 on page 225 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 Chapter 20, “Quick reference to ISMF commands and line operators,” on page 299. 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 17. Module/CLIST Names for ISMF Line Operators, Part 1

Line Operator	Aggregate Group List	Data Class	Data Set List	Optical Drive List
<b>ALTER</b>	DGTFALH1	DGTFALD1	DGTFAL01	DGTFALR1
<b>ANALYZE</b>	-	-	-	-
<b>BROWSE</b>	-	-	DGTFBR01	-
<b>CLIST</b>	-	-	DGTFCL01	-
<b>COMPRESS</b>	-	-	DGTFCM01	-
<b>CONDENSE</b>	-	-	DFQFCND1	-
<b>CONVERTV</b>	-	-	-	-
<b>COPY</b>	DGTFCAH1	DGTFCAD1	DGTFCY01	DGTFCAR1
<b>DEFRAG</b>	-	-	-	-
<b>DELETE</b>	DGTFDNH1	DGTFDND1	DGTFDL01	DGTFDNR1
<b>DISPLAY</b>	DGTFDIH1	DGTFDID1	-	DGTFDIR1
<b>DUMP</b>	-	-	DGTFDP01	-
<b>EDIT</b>	-	-	DGTFED01	-
<b>EJECT</b>	-	-	-	-
<b>ERASE</b>	DGTFDNH1	DGTFDND1	DGTFDL01	DGTFDNR1
<b>HALTERDS</b>	-	-	DFQFHA01	-
<b>HBACKDS</b>	-	-	DFQFHB01	-
<b>HBDELETE</b>	-	-	DFQFHBD1	-
<b>HDELETE</b>	-	-	DFQFHD01	-

Table 17. Module/CLIST Names for ISMF Line Operators, Part 1 (continued)

Line Operator	Aggregate Group List	Data Class	Data Set List	Optical Drive List
<b>HIDE</b>	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01
<b>HMIGRATE</b>	-	-	DFQFHM01	-
<b>HRECALL</b>	-	-	DFQFHRL1	-
<b>HRECOVER</b>	-	-	DFQFHRC1	-
<b>INIT</b>	-	-	-	-
<b>INSPECT</b>	-	-	-	-
<b>LIST</b>	-	-	-	-
<b>LISTSYS</b>	-	-	-	-
<b>LISTVOL</b>	-	-	-	-
<b>MESSAGE</b>	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00
<b>RAUTH</b>	-	-	-	-
<b>RECOVER</b>	-	-	-	-
<b>RELEASE</b>	-	-	DGTFRL01	-
<b>RESTORE</b>	-	-	DGTFRT01	-
<b>REFORMAT</b>	-	-	-	-
<b>SECURITY</b>	DGTFSRD1	-	DGTFSRD1	-
<b>SETCACHE</b>	-	-	-	-
<b>STATUS</b>	-	-	-	-
<b>TSO Commands and CLIST</b>	DGTFUS01	DGTFUS01	DGTFUS01	DGTFUS01

Table 18. Module/CLIST Names for ISMF Line Operators, Part 2

Line Operator	Optical Library List	List	Management Class	Mountable Optical Volume List	Copy Pool
<b>ALTER</b>	DGTFALL1	-	DGTFALM1	-	DGTFALP1
<b>AUDIT</b>	DGTFaul1	-	-	DGTFau01	-
<b>ANALYZE</b>	-	-	-	-	-
<b>BROWSE</b>	-	-	-	-	-
<b>CLIST</b>	-	-	-	DGTFCL01	-
<b>COMPRESS</b>	-	-	-	-	-
<b>CONDENSE</b>	-	-	-	-	-
<b>CONVERTV</b>	-	-	-	-	-
<b>COPY</b>	DGTFCAL1	-	DGTFCAM1	-	DGTFCAP1
<b>DEFRAG</b>	-	-	-	-	-
<b>DELETE</b>	DGTFDNL1	DGTFEL01	DGTFDNM1	-	DGTFDNP1
<b>DISPLAY</b>	DGTFDIL1	-	DGTFDIM1	-	DGTFDIP1



Table 18. Module/CLIST Names for ISMF Line Operators, Part 2 (continued)

Line Operator	Optical Library List	List	Management Class	Mountable Optical Volume List	Copy Pool
DUMP	-	-	-	-	-
EDIT	-	-	-	-	-
EJECT	-	-	-	DGTFEJ01	-
ERASE	DGTFDNL1	DGTFEL01	DGTFDNM1	-	-
HALTERDS	-	-	-	-	-
HBACKDS	-	-	-	-	-
HBDELETE	-	-	-	-	-
HDELETE	-	-	-	-	-
HIDE	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01
HMIGRATE	-	-	-	-	-
HRECALL	-	-	-	-	-
HRECOVER	-	-	-	-	-
INIT	-	-	-	-	-
INSPECT	-	-	-	-	-
LIST	-	DGTFLL01	-	-	-
LISTSYS	-	-	-	-	-
LISTVOL	DGTFVL1	-	-	-	-
MESSAGE	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00
RAUTH	-	-	-	-	-
RECOVER	-	-	-	DGTFRC01	-
RELEASE	-	-	-	-	-
REMAP	DGTFRML1	-	-	-	-
RESTORE	-	-	-	-	-
REFORMAT	-	-	-	-	-
SECURITY	-	-	DGTFSRD1	-	-
SETCACHE	-	-	-	-	-
STATUS	-	-	-	-	-
TSO Commands and CLIST	DGTFUS01	DGTFUS01	DGTFUS01	DGTFUS01	DGTFUS01

Table 19. 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	-	-	-	DGTFALU1	DGTFALU04

Table 19. Module/CLIST Names for ISMF Line Operators, Part 3 (continued)

Line Operator	Storage Class	Storage Group	Volume	Tape Library	Mountable Tape Volume
<b>BROWSE</b>	-	-	-	-	-
<b>BUILD</b>	-	-	DGTFBX01	-	-
<b>CLIST</b>	-	-	DGTFCL01	DGTFCL01	DGTFCL01
<b>COMPRESS</b>	-	-	DGTFCS01	-	-
<b>CONDENSE</b>	-	-	-	-	-
<b>CONSOLID</b>	-	-	DGTFCI01	-	-
<b>CONTROL</b>	-	-	DGTFCT01	-	-
<b>CONVERTV</b>	-	-	DGTFCN01	-	-
<b>COPY</b>	DGTFCAS1	DGTFCAG1	DGTFCV01	DGTFCAY1	-
<b>DEFRAG</b>	-	-	DGTFDF01	-	-
<b>DELETE</b>	DGTFDNS1	DGTFDNG1	-	DGTFDNY1	-
<b>DISPLAY</b>	DGTFDIS1	-	-	DGTFDIY1	-
<b>DUMP</b>	-	-	DGTFDM01	-	-
<b>EDIT</b>	-	-	-	-	-
<b>EJECT</b>	-	-	-	-	DGTFEJ01
<b>ERASE</b>	DGTFDNS1	DGTFDNG1	-	-	-
<b>HALTERDS</b>	-	-	-	-	-
<b>HBACKDS</b>	-	-	-	-	-
<b>HBDELETE</b>	-	-	-	-	-
<b>HDELETE</b>	-	-	-	-	-
<b>HIDE</b>	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01	DGTFHI01
<b>HMIGRATE</b>	-	-	-	-	-
<b>HRECALL</b>	-	-	-	-	-
<b>HRECOVER</b>	-	-	-	-	-
<b>INIT</b>	-	-	DGTFIN01	-	-
<b>INSPECT</b>	-	-	DGTFIV01	-	-
<b>INSTALL</b>	-	-	DGTFIL01	-	-
<b>LIST</b>	-	-	DGTFIV01	-	-
<b>LISTSYS</b>	-	DGTFLIC1	-	-	-
<b>LISTVOL</b>	-	DGTFLOC1	-	DGTFVL1	-
<b>MESSAGE</b>	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00	DGTFMS00
<b>RAUTH</b>	-	-	DGTFRA01	-	-
<b>RECOVER</b>	-	-	-	-	-
<b>REFORMAT</b>	-	-	DGTFRF01	-	-
<b>RELEASE</b>	-	-	DGTFRV01	-	-

Table 19. Module/CLIST Names for ISMF Line Operators, Part 3 (continued)

Line Operator	Storage Class	Storage Group	Volume	Tape Library	Mountable Tape Volume
<b>RESTORE</b>	-	-	DGTFRO01	-	-
<b>REVAL</b>	-	-	DGTFRB01	-	-
<b>SECURITY</b>	DGTFSRD1	-	-	DGTFSRD1	-
<b>SETCACHE</b>	-	-	DGTFCB01	-	-
<b>STATUS</b>	-	-	-	-	-
<b>TSO Commands and CLIST</b>	DGTFUS01	DGTFUS01	DGTFUS01	-	-

Table 20. Module/CLIST Names for ISMF Commands

Command	Module Name	CLIST
<b>ACTIVATE</b>	DGTFACAT	-
<b>AUDIT</b>	DGTFAU02	-
<b>BOTTOM</b>	DGTFDO01	-
<b>CLEAR</b>	DGTFCR01	-
<b>COMPRESS</b>	DGTFCP01	-
<b>COPY</b>	DGTFCO01	-
<b>DOWN</b>	DGTFDO01	-
<b>DSUTIL</b>	-	DSUTIL
<b>DUMP</b>	DGTFDU01	-
<b>ERTB</b>	DGTFER02	-
<b>FILTER</b>	DGTFFI01	-
<b>FILTER CLEAR</b>	DGTFFI01	-
<b>FIND</b>	DGTFFN01	-
<b>FOLD</b>	DGTFFU01	-
<b>LEFT</b>	DGTFLE01	-
<b>LIBRARY</b>	-	LIBRARY
<b>LISTPRT</b>	DGTFPR01	-
<b>MIGRATE</b>	-	MIGRATE
<b>PROFILE</b>	DGTFFP01	-
<b>QRETRIEV (DS)</b>	ACBFUTO3	-
<b>QRETRIEV (DVOL)</b>	ACBFUTO4	-
<b>QSAVE (DS)</b>	ACBFUTO6	-
<b>QSAVE (DVOL)</b>	ACBFUTO7	-
<b>RECALL</b>	-	RECALL
<b>RECOVER</b>	DGTFRC01	-
<b>RELEASE</b>	DGTFRE01	-
<b>RESHOW</b>	DGTFRW01	-
<b>RESTORE</b>	DGTFRR00	-

---

Table 20. Module/CLIST Names for ISMF Commands (continued)

---

Command	Module Name	CLIST
<b>RIGHT</b>	DGTFRI01	-
<b>SAVE</b>	DGTFSLDS	-
<b>SORT</b>	DGTFSO01	-
<b>TOP</b>	DGTFUP01	-
<b>UP</b>	DGTFUP01	-
<b>VALIDATE</b>	DGTFVLVA	-
<b>VIEW</b>	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 or XFACILIT class

---

In order to control the ability to perform functions associated with storage management, define profiles in the FACILITY or XFACILIT 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.

XFACILIT class is an extension of the FACILITY class that supports longer resource names. Properties listed for the FACILITY class also apply to the XFACILIT class. The following are XFACILIT profiles.

- STGADMIN.IGG.DELAUDIT.catalogname

## 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.

The RACF OPERCMDS resource class can control the ability to:

- Issue specific the VARY and DISPLAY SMS Commands.

If defined, these profiles are checked before a user is allowed to perform the protected function. The user must have READ access. 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  
STGADMIN.SMS.DISPLAY.ZHYPERLINK  
STGADMIN.SMS.VARY.ZHYPERLINK

In addition to the individual profiles, the STGADMIN.\* profile should 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 an 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. Not many users should have RACF authority to STGADMIN.DPDSRN.olddsname. Use this option only if you know that the data set is not open on any system. For more information about 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 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.IFG.READVTOC.volser**

Controls the ability to obtain READ access to the VTOC or VTOC index.

**STGADMIN.IGD.ACTIVATE.CONFIGURATION**

Controls the ability to activate an SMS configuration.

**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 to alter it. To use this profile, the administrator must have ALTER access to the data set whose storage or management class is to be changed.

**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 that is 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 or alter an NVR for a data set without affecting the BCS entry if one exists. This profile is only checked by authorized services using the LOCATE macro, not by utilities like IDCAMS.

**STGADMIN.IGG.DEFNVSAM.NONVR**

Controls the ability to define or alter a BCS entry for a data set without affecting the VVDS entry if one exists. This profile is only checked by authorized services using the LOCATE macro, not by utilities like IDCAMS.

**STGADMIN.IGG.DELETE.NOSCRATCH**

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.DELETE.RENAME**

Controls the ability to delete data set entries flagged as "rename in process". Attempts without the facility class for data sets flagged in this manner receive message IDC3009I with a return code of 90 and a reason code of 54. The "rename in progress" flag is ignored for users having RACF READ authority to the facility class and issuing a DELETE, and the entry is deleted. This facility class is intended for maintenance purposes.

**STGADMIN.IGG.DELGDG.FORCE**

Controls the ability to use DELETE FORCE on a generation data group which contains a system-managed generation data set.

If the FACILITY class entry for STGADMIN.IGG.DELGDG.FORCE exists, the user must have at least READ authority to that FACILITY class profile to perform a DELETE GDG FORCE.

If the FACILITY class entry for STGADMIN.IGG.DELGDG.FORCE exists, the user needs READ authority to the FACILITY class profile, and ALTER authority to the DATASET profile.

**STGADMIN.IGG.DELNVR.NOBCSCHK**

Controls the ability to delete the VVDS entry (the NVR) for a 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 that is 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 except for 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 main catalog to perform the DELETE VVR or DELETE NVR functions.



**Attention:** Restrict access to this FACILITY class to users who understand the risk that is 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. To do this, the user needs RACF ALTER authority to the main 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 main 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.

### **STGADMIN.SMS.ALLOW.DATASET.ENCRYPT**

Controls the ability to create encrypted data sets when the key label to be used is specified using a method other than the DFP segment in the RACF data set profile. To create (allocate the initial space for) an encrypted data set, one of the following items must be true:

- You have at least read authority to this resource in the FACILITY class.
- The DFP segment in the RACF data set profile specifies a key label. In this case, the user does not need authority to this resource in the FACILITY class.

With read authority to this resource, the user can request encryption by supplying a key label for the data set using any of the following methods:

- The DFP segment of the RACF data set profile specifies a key label.
- The DSKEYLBL keyword is coded on the JCL DD statement or the dynamic allocation equivalent.
- The SMS data class specifies a key label.
- The IDCAMS DEFINE command includes the KEYLABEL keyword.

### **STGADMIN.SMS.ALLOW.DATASET.SEQ.ENCRYPT**

Controls the ability to allow basic and large format data sets to be treated by the system as a supported data set type for data set encryption.

### **STGADMIN.SMS.DADSM.UNMAP.PREFER**

Prevents access to residual data by releasing extents and unmapping media blocks when a data set is deleted, or space is released. If the device does not support this facility, residual data is erased by overwriting with zeros. SMS-managed, non-SMS managed, and temporary data sets are supported. A data set is eligible for UNMAP when erase is requested using RACF command SETROPTS ERASE or an ERASE keyword is specified at allocation time or included as a parameter to DADSM SCRATCH or PARTREL.

Authorize access to the DADSM UNMAP facility. Define a profile in the FACILITY class:

```
RDEFINE FACILITY STGADMIN.SMS.DADSM.UNMAP.PREFER UACC(NONE)
```

Refresh the CLASS facility:

```
SETROPTS RACLIST(FACILITY) REFRESH
```

### **STGADMIN.SMS.ALLOW.PDSE.ENCRYPT**

Controls the ability to allow PDSE data sets to be treated by the system as a supported data set type for data set encryption.

### **STGADMIN.SMS.FAIL.INVALID.DSNTYPE.ENC**

Can be used to control whether an allocation should succeed or fail if a key label is specified for a DASD data set that is not extended format.

By default, the system ignores the key label and the data set is successfully created as a non-encrypted non-extended format data set.

- For an SMS-managed data sets, message IGD17156I is issued indicating that the key label is ignored.
- For a non-SMS managed data sets allocated using JCL or Dynamic Allocate, message IGD17156I is issued indicating that the key label is ignored.
- For a non-SMS managed data sets allocated using IDCAMS DEFINE, message IDC3040I is issued indicating that the key label is ignored.

When the user has at least READ authority to this resource, the system fails the allocation.

- For an SMS managed data sets, message IGD17151I is issued. For initial allocation on V2R1, message IGD17154I might be issued instead.
- For a non-SMS managed data sets allocated using IDCAMS DEFINE, message IDC3039I is issued. Using IDCAMS DEFINE on V2R1, message IDC0017I might be issued instead.



**Note:** This resource has no effect when the data set is not a DASD data set. When a key label is specified for a non-DASD data set, the key label is ignored and no message is issued.

When a key label is specified for extended format data sets, this resource is not checked. Instead, the resource STGADMIN.SMS.ALLOW.DATASET.ENCRYPT should be used to control whether a user is allowed to create non-encrypted data sets.

#### **STGADMIN.SMS.DISPLAY.ZHYPERLINK**

Allows the D SMS,DSNAME,ZHL command to be issued only if the user has read access to this resource.

#### **STGADMIN.SMS.VARY.ZHYPERLINK**

Allows the V SMS,DSNAME,ZHLWRITE=,ZHLREAD= command to be issued only if the user has read access to this resource.

#### **STGADMIN.IGG.DELAUDIT.catalogname**

Allows users with read access to this resource the ability to delete a data set cataloged in the specified catalog. *catalogname* is the name of the specified catalog appended to the resource prefix of STGADMIN.IGG.DELAUDIT. When this authority is exercised and the class is defined with the AUDIT(ALL(READ)) parameter, an SMF type 80 record is written to document this event. If the user does not have read access to the resource, the user needs ALTER authority to the data set to delete it. If the resource is not defined, users need either ALTER authority to the data set or catalog for deletion. This is behavior before the introduction of this new class. This resource class applies to all data set types including SMS, non-SMS, VSAM, and non-VSAM.

If a user issues the 'DELETE USERCATALOG FORCE' command to delete a catalog and the resource is defined for the catalog being deleted, the deletion rules described apply to the data sets within the catalog. The command deletes the catalog, however, data sets to which the user does not have authority remain and are not accessible until they are recataloged.

#### **STGADMIN.SMS.ALLOW.PDSE.COMPACT**

Controls the ability to allow PDSE data sets to be treated by the system as a supported data set type for data set compression.

#### **STGADMIN.SMS.DATASET.ENCRYPT.PREFERRED**

Controls the ability to allow an empty encrypted sequential basic format or large format data set to be converted to a non-encrypted data set when open processing detects that certain data set encryption requirements have not been met, instead of failing the open request.

- The user must have at least read authority to this resource in the FACILITY class.

See [z/OS DFSMS Using Data Sets](#) for more information about the list of restrictions to which this applies.

The following two profiles can be optionally defined to enable the functions described.

#### **STGADMIN.IGG.AUTO.BYPASS.ALLOW**

allows VSAM OPEN routines to bypass RACF security checking for programs that issue OPEN in supervisor state or protection key 0.

#### **STGADMIN.IGG.AUTO.BYPASS.LOG**

enables users to log security bypass behavior in SMF 80 records. The user must have at least READ access authority to this resource as well as to STGADMIN.IGG.AUTO.BYPASS.ALLOW. See [RACF protection for VSAM data sets in z/OS DFSMS Using Data Sets](#).

## **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. For audit purposes, IBM recommends logging all access to the STGADMIN.IGD.ACTIVATE.CONFIGURATION resource in the FACILITY class by specifying either AUDIT(ALL) or GAUDIT(ALL).

For more information on DFSMSdss functions, see *z/OS DFSMSdss Storage Administration*.

For listing of the Storage Administrator FACILITY Class Profiles for DFSMSdss, see *z/OS DFSMSdss Storage Administration*.

## 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
- Data set key label, DATAKEY, 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/E and ISPF

To define an SMS data set under TSO/E 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/E, 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 DATAKEY 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 18. 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.

**Note:** VSAM RLS requires that the data sets be System Managed Storage (SMS) data sets. To be eligible for RLS, a data set that is not already SMS-managed must be converted to SMS.

This topic 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” on page 235](#)
- [“Defining CF cache structures in the SMS base configuration” on page 250](#)
- [“Defining storage classes for VSAM RLS” on page 251](#)
- [“Activating VSAM RLS” on page 252](#)
- [“Monitoring the coupling facility for VSAM RLS” on page 253](#)
- [“Recovering VSAM RLS processing” on page 260](#)
- [“Falling back from VSAM RLS processing” on page 262](#)

See [z/OS DFSMS Using Data Sets](#) for more information on VSAM RLS. For information on the online Parallel Sysplex Configuration Assistant, see [IBM z/OS Parallel Sysplex \(www.ibm.com/products/zos/parallel-sysplex\)](#).

---

### 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” on page 235](#)
- [“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 242](#)
- [“Defining the primary CF lock structure” on page 244](#)
- [“Modifying the SYS1.PARMLIB IGDSMSxx member” on page 248](#)
- [“Establishing authorization for VSAM RLS” on page 249](#)

### 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 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 [\*z/OS Upgrade Workflow\*](#) 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, because 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.

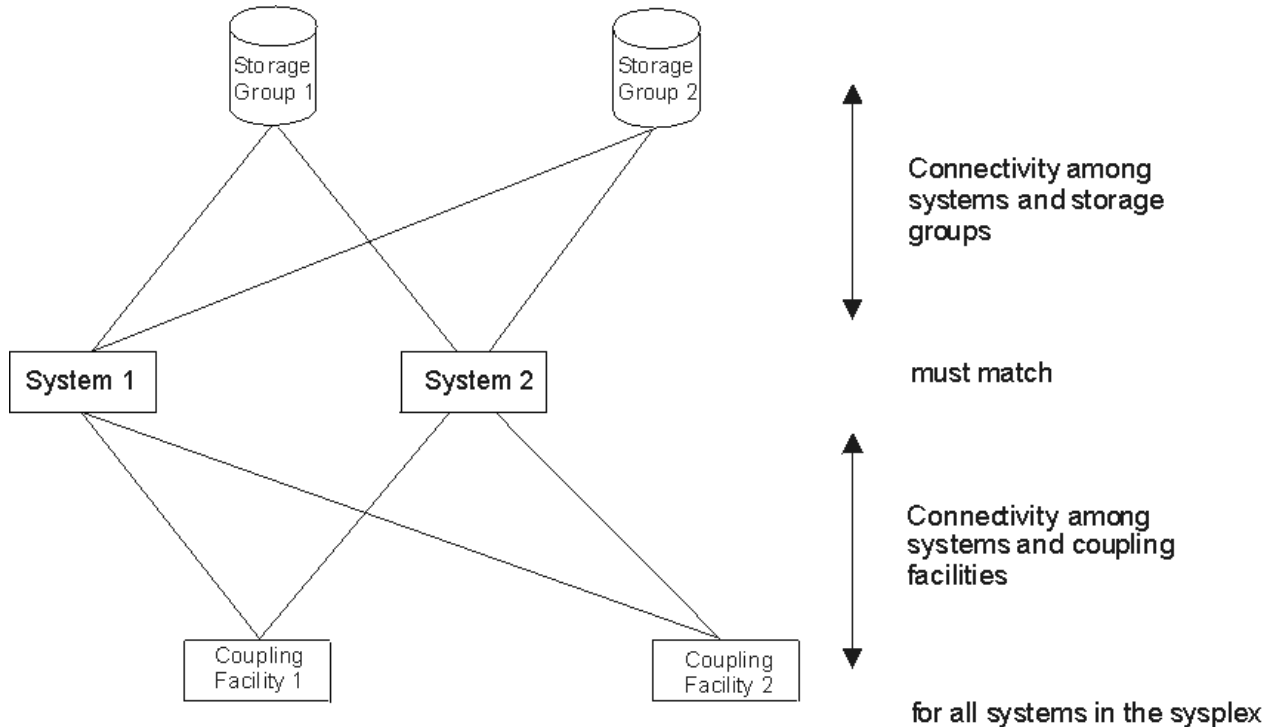
For recoverable data set, in addition to the data sharing across CICS applications, VSAM RLS enables read-with-integrity sharing by batch jobs. Batch jobs can share recoverable files while they are being modified by CICS applications. This is possible because VSAM provides the record locking and buffer coherency functions across CICS and batch. Because 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, because 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.

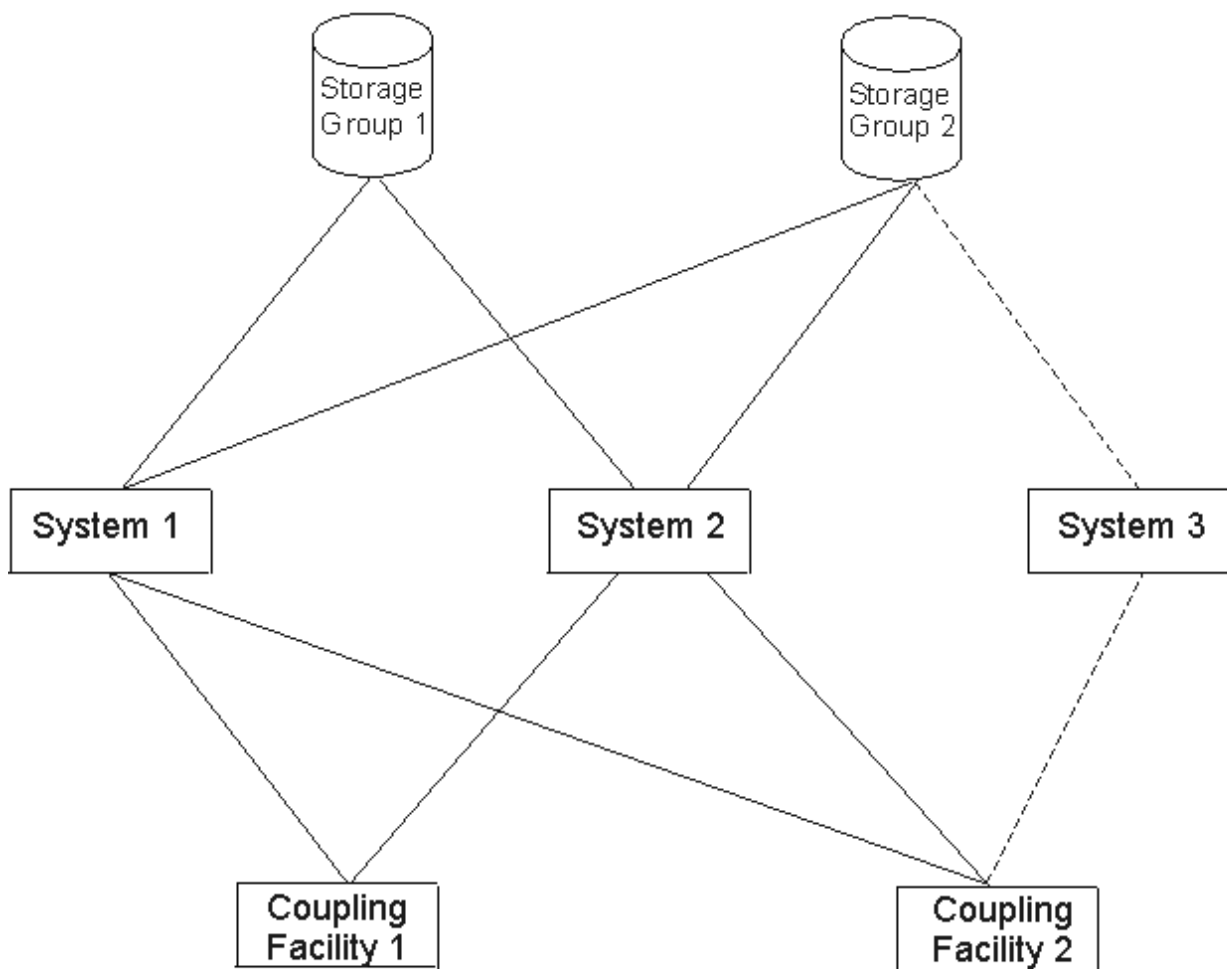
Figure 43 on page 237 shows how connectivity among systems and storage groups matches connectivity among systems and coupling facilities.



DA6S2003

Figure 43. Example of Global Connectivity in a Parallel Sysplex

Figure 44 on page 238 illustrates the problems that can occur if you do not have global connectivity across the Parallel Sysplex:



DA6S2004

Figure 44. Example of Insufficient Connectivity in a Parallel Sysplex

In Figure 44 on page 238, 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. Because 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 this 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.

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 might 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 data can be either SMS or non-SMS managed.

The SHCDS acts as a log for sharing support. It is a logically partitioned linear data set, with CISIZE of 4096, that must 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 SHCDS. Place these data sets for maximum availability. If necessary, you can have up to five active SHCDSs and five spare SHCDSs.

**Restriction:** Because the contents of these data sets are highly dynamic, do not use backup and restore functions because these might cause the loss of VSAM RLS data set recovery information.

Consider the following as you allocate and maintain your SHCDSs:

- The SHCDS must not be shared outside of the Parallel Sysplex. Refer to the restrictions in [“Running SMS in a parallel sysplex environment”](#) on page 6.
- Allocate SHCDSs so that the number of active and spare data sets ensures that 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 because 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.
- 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.
- Ensure that the space allocation for active and spare SHCDSs is the same. Ensure that data sets have enough space for growth. Ensure that the SHCDS has secondary extents defined.

- 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.Vvolser
```

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.

Define the SHCDS with the following characteristics:

- CISIZE equals 4096
- Shareoptions are (3,3)
- Contains secondary extents
- Resides on a single volume

Use the following formula to calculate the size of the primary extent of your SHCDS:

```
space = (number_of_recoverable_data_sets * number_of_recoverable_subsystems * 2 * 272) bytes
        or 13MB, whichever is larger
```

where:

***number\_of\_recoverable\_data\_sets***

Is the average number of recoverable data sets open across the sysplex.

***number\_of\_recoverable\_subsystems***

Is the average number of recoverable subsystems that are registered with RLS across the sysplex.

For example, if you have an average of 10 recoverable subsystems that are registered with RLS with an average of 1000 recoverable data sets open, the amount of space that is required for the primary extent is (10 \* 1000 \* 2 \* 272) or 5440000 bytes (5313 KB).

You can create an SHCDS using access method services. 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. Specify CISIZE as 4096. Select a volume that is a member of a storage class with the guaranteed space attribute.

Figure 45 on page 240 shows how to create an SHCDS using IDCAMS:

```
/*-----
/* ALLOCATE SHCDS ON XP0301 - SXPXS04 IS GUARANTEED SPACE
/*-----
//ALLOCLD1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE CLUSTER (NAME(SYS1.DFPSHCDS.ACTIVE.VXP0301)
    LINEAR
    CISZ(4096)
    STORCLAS(SXPXS04)
    SHAREOPTIONS(3 3)
    CYL(10 5)
    VOLUME(XP0301) )
/*
```

Figure 45. Example for creating sharing control data sets using IDCAMS

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 fully qualified SHCDS name. For example, VARY SMS, SHCDS (ACTIVE.XP0301). 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 through subsequent IPLs.

The names of the SHCDS data sets are stored in the sysplex couple data sets. In the event of starting up with a new set of sysplex couple data sets, the SHCDS data sets names are lost. To make them available, use the VARY SMS SHCDS command. .

To delete and redefine an SHCDS data set, you must first delete the SHCDS data set from SMSVSAM by entering the V SMS.SHCDS(*shcdsname*).DELETE command, and then you can use IDCAMS DELETE and DEFINE to modify the new SHCDS data set. You must do that even if SMSVSAM is not active. Note, that you always must have at least two active SHCDS data sets and one spare SHCDS data set. To delete an SHCDS data set when there are only the minimum number of SHCDS data sets defined, you must first define additional SHCDS data sets before you can delete the existing SHCDS data sets.

The following example shows how to delete and redefine SHCDS data sets when only the minimum number of SHCDS data sets are defined:

1. If the following SHCDS data sets are already defined:

```
16.40.09 SYSTEM          d sms,shcds
IGW612I 16:40:10    DISPLAY SMS,SHCDS
Name              Size      %UTIL   Status   Type
TOOSMALL.VXP0301   7200Kb    5%     GOOD    ACTIVE
TOOSMALL.VXP0302   7200Kb    5%     GOOD    ACTIVE
WRONGVOL.VXP0201   7200Kb    5%     GOOD    SPARE
```

2. Enter these commands to define three new data sets:

```
V SMS,SHCDS(JUSTRITE.VXP0301),NEW
V SMS,SHCDS(JUSTRITE.VXP0302),NEW
V SMS,SHCDS(RIGHTVOL.VXP0202),NEWSPARE
```

3. These are the resulting SHCDS data sets:

```
SYSTEM1              d sms,shcds
IGW612I 16:45:49    DISPLAY SMS,SHCDS
Name              Size      %UTIL   Status   Type
TOOSMALL.VXP0301   7200Kb    5%     GOOD    ACTIVE
TOOSMALL.VXP0302   7200Kb    5%     GOOD    ACTIVE
JUSTRITE.VXP0301   14400Kb   2%     GOOD    ACTIVE
JUSTRITE.VXP0302   14400Kb   2%     GOOD    ACTIVE
WRONGVOL.VXP0201   7200Kb    5%     GOOD    SPARE
RIGHTVOL.VXP0202   7200Kb    5%     GOOD    SPARE
```

4. Enter these commands to delete the old data sets:

```
V SMS,SHCDS(TOOSMALL.VXP0301),DELETE
V SMS,SHCDS(TOOSMALL.VXP0302),DELETE
V SMS,SHCDS(WRONGVOL.VXP0201),DELETE
```

5. The resulting SHCDS data sets are as follows:

```
SYSTEM1              d sms,shcds
IGW612I 16:45:49    DISPLAY SMS,SHCDS
Name              Size      %UTIL   Status   Type
JUSTRITE.VXP0301   14400Kb   2%     GOOD    ACTIVE
JUSTRITE.VXP0302   14400Kb   2%     GOOD    ACTIVE
RIGHTVOL.VXP0202   7200Kb    5%     GOOD    SPARE
```

**Note:**

1. If you swap in a new set of SHCDS data sets or make changes to an existing SHCDS data set, you must communicate the changes to SMSVSAM through the VARY SMS SHCDS command.
2. Do not move an SHCDS data set from one volume to another. The SHCDS data set naming convention depends on the volume in which the SHCDS data set resides.
3. During disaster recovery, when the RLS SMSVSAM address space is initialized, you can reuse an existing active SHCDS data set without losing any information that was stored in it. If you decide that, in case of disaster recovery, the RLS SAM recovery information from the previous instance of the

sysplex will need to be retained for the reinitialization of the RLS SMSVSAM address space, you can make an existing active SHCDS available for reuse during disaster recovery. In that case, issue:

```
RLS SMSVSAM address space, V SMS,SHCDS(shcdsname),OLD
```

For the initializing RLS SMSVSAM address space, the reused SHCDS must be the first one activated with the OLD command. Subsequent active SHCDSs will be copied from the first active SHCDS.

For more information see [“Changing the state of coupling facility cache structures and volumes” on page 259.](#)

## 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. If additional space is required, RLS manages the altering of the cache, up to the maximum size specified. Do not specify ALLOWAUTOALT(YES) for RLS cache structures, which would allow system-initiated alters, thus preventing RLS from being given control to manage the 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.

## Sharing CF structures

You can share CF structures among applications. However, this results in all data being treated equally, which may not be desirable. If you want to treat all your RLS data equally, sharing CF structures might be an appropriate and relatively simple approach. However, if you want to give some data sets preference over other data sets, you might want to use several CF structures of different sizes. For example, you might place test data sets in a small structure and production data sets in a large structure. Or, to give CICS data preference over DFSMSHsm data, place the CICS data in a larger structure and DFSMSHsm data in a smaller structure.

## 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
<b>FOR_1</b>	20 MB	30 MB	50 MB
<b>FOR_2</b>	40 MB	no pool	40 MB
<b>FOR_3</b>	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.

## 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 pools. 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.

If RLS\_Max\_Pool\_Size is greater than 1500M, SMS will pass a value of 9999. The value will then be reset to 1728M. A default upper limit is set and this internal upper limit can be changed.

**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
<b>FOR_1</b>	20 MB	30 MB	50 MB	75 MB
<b>FOR_2</b>	40 MB	no pool	40 MB	60 MB
<b>FOR_3</b>	30 MB	50 MB	80 MB	120 MB

If you have multiple applications using RLS, the RLS\_MAX\_POOL\_SIZE value needs to take into account the requirements for all of the applications.

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 primary CF lock structure

**Requirements:** To use VSAM RLS, you must define a single, master CF lock structure. If, however, you use the DUPLEX attribute, you only define one CF lock structure. That is because the structure gets created automatically into the secondary CF.

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 (DUPLEX (ENABLED) or DUPLEX (ALLOWED) in the CFRM policy.

For more information about system-managed lock structure duplexing, see [“Recovering the CF lock structure”](#) on page 260.

The Primary CF lock structure is named IGWLOCK00. Use the XCF 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 21. 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 22 on page 244 shows some sample lock allocation estimates:

*Table 22. CF Lock Structure Sizing Examples*

MAXSYSTEM Value	Number of systems	Total Lock Size
<= 7	2	64MB
	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.

**Note:** XES rounds up the number of lock entries to the next highest power of two. For example, if you base your calculations on  $2 * 10M = 20M$  locks, XES allocates the structure with 32M locks.

The primary DFSMS CF lock structure, IGWLOCK00, has persistent connections and is a persistent structure. To delete this structure you must use the operator command:

```
V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE
```

## 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, warning message IGW326W 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.

Because 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 by 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. Because 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 21 on page 244](#).



## 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:

$$\text{Minimum Lock Structure Size} = F * M / T$$

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:

$$\text{Initial Lock Structure Size} = 10M * 2 * 2$$

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:

$$\text{Minimum Lock Structure Size} = 1.5 * 32MB / 0.5$$

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, because 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, if YES, specifies whether the SMSVSAM address space is started as part of system initialization or by the V SMS,SMSVSAM,ACTIVE command. If you specify NO, to start SSVSAM later, you must do the following: change RLSINIT to YES, issue the SET SMS=xx command, then IPL or issue the V SMS,SMSVSAM,ACTIVE command. The default is NO.
- 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
- RlsAboveTheBarMaxPoolSize, which specifies the maximum amount of virtual storage above the 2-gigabyte bar that can be used for VSAM RLS buffering.
- RlsFixedPoolSize, which specifies the amount of real storage (both above and below the 2-gigabyte bar) to be dedicated to VSAM RLS buffering.
- SMF\_TIME, which aligns the SMF type 42 records for DFSMS (with subtypes 2, 15, 16, 17, 18 and 19) to the SMF\_TIME interval
- USEEAV, which specifies at the system level, whether SMS is to select extended address volumes during volume selection processing. This check applies to new allocations and when extending data sets to a new volume. You can enter an operator command to change the value of this keyword.

### NO

This is the default. SMS does not select extended address volumes during volume selection.

Setting USEEAV to NO has the following implications:

- Data sets might still exist on extended address volumes in either the track-managed or cylinder-managed space of the volume.
- In addition, when you specify or default to USEEAV=NO, the system skips EAV volumes for the following functions to avoid the failure of DFSMSshm requests:
  - Recall
  - Migration to an ML1 or ML2 DASD volume
  - Backup
  - Recover
  - ARECOVER
  - Daily backup volume / backup volume spill process
- When a system where USEEAV is set to NO backs up a data set from an EAV, the data set will be recovered to a non-EAV volume and the vendor attributes will be lost. The system issues message ARC0784I in this case.

### YES

SMS is to use extended address volumes to allocate new data sets or to extend existing data sets to new volumes.

**Note:** When SMS is not active, USEEAV is not available and the installation must use alternate means to control the usage of EAVs.

- BreakPointValue, which specifies the number of cylinders (0-65520) that SMS uses to select volumes for VSAM data sets. The default is 10. When SMS is not active, the system assigns the default value.

You can modify these parameters at any time during VSAM RLS processing. For more information, refer to [“Changing IGDSMSxx parameters to support the coupling facility” on page 256](#). For more information

about the IGDSMSxx member, refer to the topic about [IGDSMSxx](#) in *z/OS MVS Initialization and Tuning Reference*.

## Establishing authorization for VSAM RLS

To establish authorization to access VSAM RLS resources, assign a RACF attribute of PRIVILEGED or TRUSTED to the VSAM RLS server address space, SMSVSAM.

With PRIVILEGED, most RACROUTE REQUEST=AUTH macro instructions that are done for SMSVSAM are considered successful, without any checking being performed. The checking that is done for the CHKAUTH operand on the RACROUTE REQUEST=DEFINE macro instruction is also bypassed. All other RACF processing occurs as usual. RACF does not:

- Call any exit routines.
- Generate any SMF records.
- Update any statistics.

TRUSTED is similar to PRIVILEGED. Most RACROUTE REQUEST=AUTH macro instructions that are done for SMSVSAM are considered successful, without any checking being performed. RACF does not:

- Call any exit routines.
- Update any statistics.

RACF does generate SMF records that are based on the audit options specified in SETROPTS LOGOPTIONS and the UAUDIT setting in the USER ID profile.

If the VSAM RLS server address space is not PRIVILEGED or TRUSTED, grant the SMSVSAM server the appropriate access authorization:

1. Add SMSVSAM with the STARTED attribute if you are using a started task group.
2. Authorize SMSVSAM for update access to SYS1.DFPSHCDS.\* data sets. If you protect SYS1.\* data sets be sure that SMSVSAM is able to access SYS1.DFPSHCDS.\* for update.
3. If you protect volumes that contain RLS-accessed data then authorize SMSVSAM for update access to the volume profiles.
4. To use the access method services SHCDS command, you must be authorized to the STGADMIN.IGWSHCDS.REPAIR resource in the FACILITY class. The SHCDS command lists SMSVSAM recovery that is associated with subsystems and spheres, and controls that recovery.
5. If using a third-party vendor security product to open data sets defined with the KEYLABEL attribute (encrypted) and to perform data set encryption/decryption, you might need to authorize SMSVSAM for read access to the CSFKEYS resource class.

Ensure that only those users who need the capability, such as CICS subsystems, 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 z/OS \(www.ibm.com/docs/en/cics-ts\)](http://www.ibm.com/docs/en/cics-ts).

## Using dssTimeOut

Specifies the number of seconds that DFMSMSdss will wait during backup processing for quiesce data set requests to complete. Specify a value from zero to 65536 seconds (which is more than 18 hours). If you specify a value between 1 and 299 seconds, the system uses a value of 300 seconds (which equals 5 minutes).

The value specified in the DSSTIMEOUT parameter value is activated when the first instance of the SMSVSAM address becomes active in the sysplex. All subsequent SMSVSAM instances will use the same value.

You can alter the DSSTIMEOUT value dynamically in the following ways:

- Using the SETSMS DSSTIMEOUT(nnnn) command
- By adding or updating the DSSTIMEOUT parameter in IGDSMSxx parmlib member and then activating it with the SET SMS=xx command

The default is 0.

## 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.

Refer to [“Defining CF cache structures” on page 242](#) and [“Defining the primary CF lock structure” on page 244](#) for more information on using CFRM policies to define CF cache and lock structures.

Perform the following steps to define CF cache structures to DFSMS.

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

- 
2. Supply values on the CDS Application Selection panel:

**CDS Name**

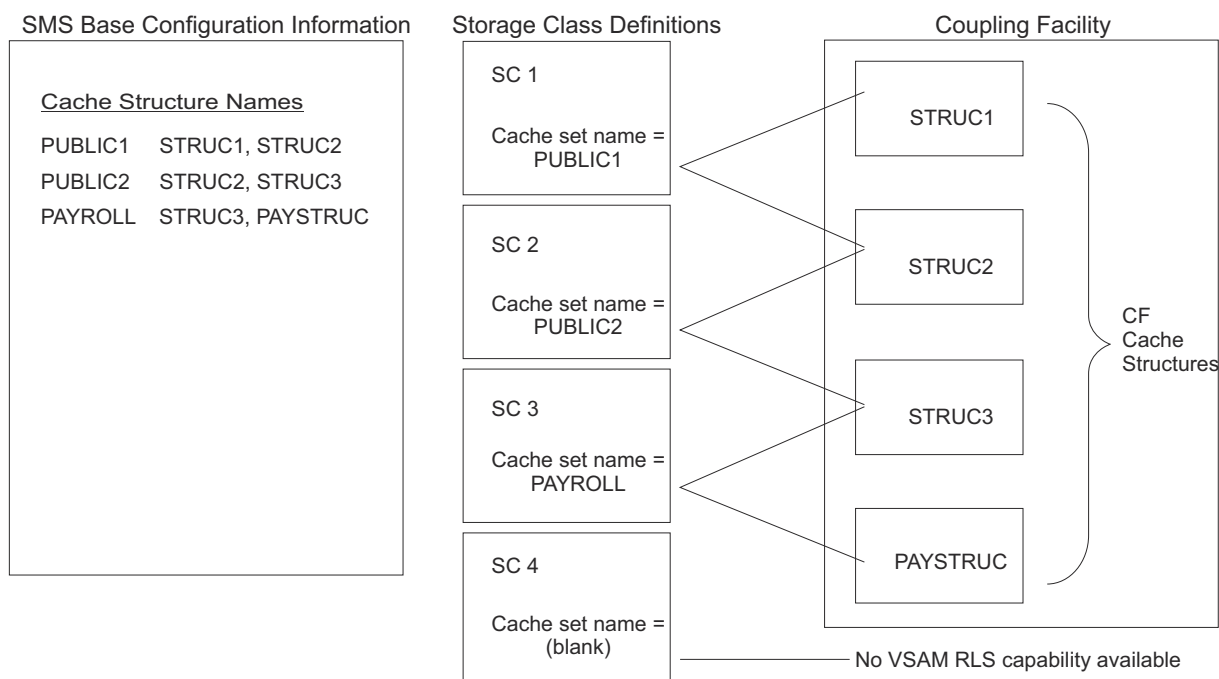
Specify the name of the SCDS that is to contain the base configuration for VSAM RLS. The CDS Name field is primed with the last used SCDS name. This name might differ from the SCDS name of your base configuration for VSAM RLS.

**Option**

Select option 7, Cache Update. This displays the CF Cache Set Update panel.

- 
3. Define your CF cache sets on the CF Cache Set Update panel. The 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 46 on page 251 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.



DA6S2005

Figure 46. 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, perform these steps.

1. Select option 5, Storage Class, from the ISMF Primary Option Menu for Storage Administrators. This displays the Storage Class Application Selection panel.

- 
2. Supply values on the Storage Class Application Selection panel:

**CDS Name**

Is the name of the SCDS you defined for VSAM RLS in the CDS Name field.

**Option**

Select option 3, Define. This displays the Storage Class Define panel.

- 
3. Press the DOWN key to view the second page of the Storage Class Define panel.

- 
4. Supply values on the Storage Class Define panel:

**CF Cache Set Name**

Is the name of the CF cache set you defined in the base configuration.

**CF Direct Weight or CF Sequential Weight**

Specify a weight attribute for the data in one of the 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 118](#) 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.

SYSVSAM must not be in the RNLDEF exclude list for any data sets accessed by RLS.

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 in [Figure 47 on page 253](#).

```
DISPLAY SMS,SMSVSAM[,ALL]
           ,CFCACHE(CF_cache_structure_name|*)
           ,CFLS
           ,CFVOL(volume_serial_number)
           ,MONDS(specification_mask|*)
           ,QUIESCE
           ,SEP
           ,SHCDS
```

*Figure 47. DISPLAY SMS command*

The following list describes the various forms of the DISPLAY SMS command and what information is displayed:

- `DISPLAY SMS,SMSVSAM`  
displays the status of lock structures connected to the system from which the command is entered.
- `DISPLAY SMS,SMSVSAM,ALL`  
displays the status of all lock structures in the sysplex. If the lock structure is duplexed, composite state information is also 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 242](#) 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  
RlsAboveTheBarMaxPoolSize  
RlsFixedPoolSize  
SMF\_TIME

You can change the values for these parameters during VSAM RLS processing only when all the spheres are closed. 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|AMx|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.

With feature level A the user can also specify Mx values in the form of AMx to customize IGW500I messages for open spheres as follows:

**M0**

IGW500I messages for all open spheres will be suppressed.

**M1**

IGW500I messages for all open spheres will be displayed.

**M2**

Only for the first sphere opened in the sysplex will an IGW500I message be displayed.

**M3**

IGW500I messages for all open spheres in the sysplex will be sent to hardcopy.

**M4**

Only for the first sphere opened in the sysplex will an IGW500I message be sent to hardcopy.

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:** The restrictions are described below.

- If A is specified for the RLS\_MaxCfFeatureLevel parameter, systems lower than z/OS 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.
- The Mx modifier can only be combined with feature level A in the form of AMx, otherwise an IGW467I will be issued to indicate that the Parmlib value specified is not valid.
- If the SETSMS command is used to change the RLS\_MaxCfFeatureLevel value to Z when there are VSAM spheres assigned to cache structures, 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**.

#### **[RlsAboveTheBarMaxPoolSize{( sysname, maxrls; ...) | (ALL, maxrls)}]**

Specifies the maximum amount of virtual storage above the 2-gigabyte bar that can be used for VSAM RLS buffering in a system in the sysplex, where:

*sysname* identifies a system and *maxrls* is specified in megabytes for that system. Multiple sets of *sysname, maxrls* pairs may be specified, separated by semicolons.

(ALL, *maxrls*) specifies that each system in the sysplex will have the same specified value.

The default for *maxrls* is 0, and the specifiable values are 500 to 2000000 (that is, 500MB to 2 terabytes).

#### **[RlsFixedPoolSize{( sysname, maxrls; ...) | (ALL, maxrls)}]**

Specifies the total amount of real storage (both above and below the 2-gigabyte bar) to be dedicated to VSAM RLS buffering in a system in the sysplex, where:

*sysname* identifies a system and *maxrls* is specified in megabytes for that system. Multiple sets of *sysname, maxrls* pairs may be specified, separated by semicolons.

(ALL, *maxrls*) specifies that each system in the sysplex will have the same specified value.

The default for *maxrls* is 0, and the specifiable values are from 0 to 80% of the total unpinned available real storage. When 80% is reached, the RLS address-space SMSVSAM Initialization will issue a message to warn of the limit.

#### **[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 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: 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 sets 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. For example, if you specify \* or \*\*, it must be the last qualifier, that is, you would code xxx.\* or xxx.\*\*

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.

For every sphere monitored, an internal control block is created in the MMF data space for each component of that sphere. Components include the data and index components of an open data set, as well as those of any alternate indexes defined to that data set. If too many spheres are monitored at once, an MMF storage pool fills up and runs out of extents. This leads to an ABEND0F4 RC24 RSN2541E0C

when another monitor request is issued, and MMF is disabled until SMSVSAM is recycled. The exact limitations are determined by storage availability, but in a normal environment the pool is able to handle approximately 200,000 total components monitored.

Monitoring a large number of data sets can have performance implications. It is recommended that only essential data sets be monitored.

## 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 structure reverts to simplex mode, and continues to operate as it would in user-managed mode (see [“User-managed lock structure rebuild” on page 260.](#))

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 specifying 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.

For more detailed information about the system-managed duplexing rebuild function, refer to [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 sharing 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 and restore procedures for SHCDSs are ineffective. An existing SHCDS can be specified to be reused "as is" in an initialization of the RLS SMSVSAM address space.

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 reuse an already-existing active SHCDS during initialization:

```
VARY SMS,SHCDS(SHCDS_name),OLD
```

- To delete either an active or a spare SHCDS:

```
VARY SMS,SHCDS(SHCDS_name),DELETE
```

- In the OLD command, the first SHCDS specified must be one of the previously active SHCDSs. The OLD command will copy the first OLD SHCDS to subsequent SHCDSs added.

The OLD command should only be issued when initializing a parallel sysplex with new, previously unused XCF CFRM couple data sets to replace any existing damaged CFRM couple data sets. The new XCF couple data sets will not contain the names of the active and spare SHCDSs that the sysplex may have been using, which will prompt SMSVSAM initialization to issue message IGW611A. Initialization of SMSVSAM will be suspended until SHCDSs are provided by the OLD or NEW and NEWSPARE commands. Issue an OLD command to reuse an existing active SHCDS only if the existing recovery information needs to be preserved; otherwise, issue the NEW command to use a new active SHCDS.

## 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 (but any secondary lock structures are not deleted), 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 LISTSUBSYS 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 Commands](#) 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.

Enter 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. Enter the following MVS command to disable the SMSVSAM server:

```
*ROUTE ALL V SMS,SMSVSAM,TERMINATESERVER
```

Reply C to any outstanding IGW418D message. If FORCE SMSVSAM,ARM does not disable the SMSVSAM server, use FORCE SMSVSAM.

- 
9. Enter 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.

**Note:** This command deletes the IGWLOCK00 lock structure, but does not delete any secondary lock structures. To delete secondary lock structures, use the SETXCF FORCE DELETE command. To completely fall back, you must also remove all secondary lock structures from the CFRM policy.

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.

## Terminating the SMSVSAM address space

---

To terminate the SMSVSAM address space, enter the following command from the MVS console:

```
V SMS,SMSVSAM,TERMINATESERVER
```

**Note:** Use the VSMS,SMSVSAM,TERMINATESERVER command before you partition a system out of the XCF Sysplex. Failure to do so can result in unexpected abends in the SMSVSAM address space.

## Quiescing or enabling a secondary lock structure

---

To quiesce (stop) or enable a VSAM RLS secondary lock structure, enter the VARY SMS command, as follows:

```
V SMS,CFLS(lockstructurename),Quiesce|Enable
```

When a VSAM RLS secondary lock structure is enabled, SMSVSAM attempts to connect to the structure during open processing for a VSAM data set using a storage class containing a lock set that references that lock structure. Once the data set is open, the lock structure holds record locks for the sphere.

The quiesce option stops access to the VSAM RLS secondary lock structure. SMSVSAM does not allow any new spheres to access this lock structure. All spheres that are already connected are allowed access until they close. When all of the connected spheres close, the lock structure transitions from Quiescing to Quiesced and SMSVSAM disconnects from the lock structure.

**Note:** The VARY SMS command is not valid for the primary CF lock structure IGWLOCK00.

## Deleting a VSAM RLS lock structure

---

You can delete secondary lock structures only with the operator command. All DFSMS lock structures are persistent structures that have persistent connections. When the secondary lock structure transitions to Quiesced state, SMSVSAM does not unallocate the lock structure in the coupling facility.

If you want to delete a secondary lock structure, the SMSVSAM address space previously connected must be terminated. You can use the following command:

```
V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE(lockstructurename)
```

To delete the primary lock structure, IGWLOCK00, all SMSVSAM address spaces must be terminated. You can use the following commands:

```
V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE(lockstructurename)
```

or

```
V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE
```

For more information, see [“Terminating the SMSVSAM address space”](#) on page 264.

## Displaying information about a secondary lock structure

---

To display information about one secondary lock structure or about all secondary lock structures, use the DISPLAY SMS command, as follows:

```
D SMS,CFLS(ALL | lockstructurename)
```

You can use the D SMS,SMSVSAM operator command to display the status of multiple SMSVSAM addresses spaces and secondary lock structures.

To display the status of the lock structures connected to the system from which you issued the command, use:

```
D SMS,SMSVSAM
```

To display the status of all the lock structures in the sysplex and the number of spheres currently associated with each, use:

```
D SMS,SMSVSAM,ALL
```



---

## Chapter 19. Writing ACS routines

This topic documents intended Programming Interfaces that allow you to write programs to obtain the services of DFSMS. This topic 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 topic 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 270.

#### 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 270. 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 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 290](#) 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, &RECOR, 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 290 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 292.

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 quotation marks and separated by commas (for example, 'SG1', 'SG2').

Table 23 on page 269 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 23. 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.



**Attention:** In the data class ACS routine, the &DSNTYPE, &DSORG, &MAXSIZE, &NVOL, &RECOR, and &SIZE variables all default to null if no corresponding value is specified in the JCL. Some values of the &DSNTYPE variable are set from values on the DD statement or dynamic allocation.

All of the read-only variables appear in [Table 24 on page 271](#). The read-only variables *are* case-sensitive. The following pages explain the uses of the read-only variables.



Table 24. Read-Only Variables

Variable	Variable	Variable	Variable	Variable
&ACCT_JOB	&DEF_MGMTCLAS	&GROUP	&MEMNQUAL	&SECLABL
&ACCT_STEP	&DEF_STORCLAS	&HLQ	&MSPDEST	&SECOND_QTY
&ACSENVIR	&DSKEYLBL	&JOB	&MSPARM	&SIZE
&ACSENVIR2	&DSN	&LABEL	&MSPOLICY	&SPACE_TYPE
&ALLVOL	&DSNTYPE	&LIBNAME	&MSPOOL	&STEP
&ANYVOL	&DSN_VERSION	&LLQ	&NQUAL	&SYSNAME
&APPLIC	&DSORG	&MAXGENS	&NVOL	&SYSPLEX
&BLKSIZE	&DSOWNER	&MAXSIZE	&PGM	&TIME
&DATE	&DSTYPE	&MEMHLQ	&RECORG	&UNIT
&DAYOFWEEK	&EATTR	&MEMLLQ	&RETPD	&USER
	&EXPDT	&MEMN		&USER_ACSVAR
&DB2SSID	&FILENUM			&XMODE
&DD				
&DEF_DATACLAS				

## 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 286.](#))

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 286.](#))

Type: Literal

Max value: 142 characters

### &ACSENVIR

The environment in which the ACS routine was invoked, one of:

#### ALLOC

for new data set allocations (this is the default)

#### ALLOCTST

for a pre-allocation or allocation test environment.

#### CHANGE

OSREQ object change environment

#### CONVERT

for data set convert in place operations

#### CTRANS

OSMC object class transition environment

#### 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

**SPMGCLTR**

DFSMSHsm class transition environment

**STORE**

OSREQ object store environment

**other**

installation exit can set its own value before re-invoking ACS

Type: Literal

Max value: 8 characters

**&ACSENV2**

The subenvironment in which the ACS routine was invoked. The valid values are the values for &ACSENVIR as well as FLASHCPY and blank.

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 286](#) 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 that are 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.



**Attention:** &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 286](#) 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 that is 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

**&DATE**

The current date in which the ACS routine was invoked. Format in the form of YYYYDDD where YYYY is the year and DDD is the day of year within the range of 1900001 to 2155366 (accounting for leap year).

Type: Literal

Max value: 7 characters, '2155366'

Using the &DATE read-only within a comparison has additional checks to be considered valid. When &DATE is compared within a statement, the literal string must be within the acceptable range where YYYY is between 1900-2155 and DDD between 001-366 (accounting for leap year).

For example,

IF &DATE > 2020000 THEN ... is invalid

IF &DATE >= 2020001 THEN ... is valid

**&DAYOFWEEK**

The current day of the week based on the local time zone.

Type: Literal

Valid Values: 'SUNDAY', 'MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY', 'SATURDAY'

Using &DAYOFWEEK within a comparison has additional checks to be considered valid. When &DAYOFWEEK is compared within a statement, the literal string must be one of the acceptable days of the week.

For example,

IF &DAYOFWEEK = 'TUES' THEN... is invalid

IF &DAYOFWEEK = 'TUESDAY' THEN... is valid

**&DB2SSID**

The SSID of the Db2 subsystem associated with an instance of OAM.

Type: Literal

Max value: 4 characters

**&DD**

DDNAME in the DD statement of the data set.

Type: Literal

Max value: 8 characters

**&DEF\_DATACLAS**

The data class name that is 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 that is 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 that is 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

#### **&DSKEYLBL**

The data set key label whose value is from the following precedence orders:

- From RACF DFP data set profile
- From JCL, IDCAMS DEFINE, dynamic allocation
- From data class

Type: Literal

Max value: 64 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 names 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. 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:

##### **BASIC**

The data set is not extended format or large format.

##### **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

##### **LARGE**

Large format data set

##### **LIBRARY**

PDSE, partitioned data set extended

##### **PDS**

Partitioned data set

##### **null**

No value specified

Type: Literal

Max value: 7 characters

**Note:** Some values of the &DSNTYPE variable (BASIC, EXC, EXR, and LARGE) are set from values on the DD statement or dynamic allocation, as well as from the data class or the LIKE parameter.

#### **&DSN\_VERSION**

The data set version number. Only supported for PDS/E version 1 or 2 and Extended Format Required or Preferred data types of version 1 or 2. Default value is 0.

Type: Numeric

Valid Values: 0, 1, or 2

#### **&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

**Restriction:** When DIRECTORY blocks are specified on the SPACE parameter in the JCL, &DSORG is set to PO.

#### **&DSOWNER**

The name of the user or group that owns the data set (set only if RACF is installed). For more details, see [“Identifying the resource owner and extracting the default classes” on page 217.](#)

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

#### **&EATTR**

EATTR (extended attributes) value, one of:

**OPT**

Extended attributes are optional

**NO**

The data set cannot have extended attributes

**blank**

Not specified. This is the default.

Type: Literal

Max value: 4 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: 5 characters

**&GROUP**

The RACF defined default group associated with the user, or the group specified in the GROUP keyword on the JCL JOB statement. If the environment is recall or recover, DFSMSHsm sets &GROUP for both authorized and non-authorized users if it is available. Therefore, do not rely on a null value for &GROUP to determine if a user is a DFSMSHsm Authorized User or not. 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 288](#) 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

#### **&MAXGENS**

The maximum number of generations for members in a PDSE. Nonzero values are only supported for PDS/E version 2 or higher.

Type: Numeric

Max Value: 2000000000

#### **&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 primary + (123 \* secondary \* volume count) if extent constraint removal is set to No. If extent constraint removal is set to Yes, the value is &SIZE plus 7257 extents. See [“Using read-only variables” on page 281](#) for more information about the values of &MAXSIZE and &SIZE for VSAM data sets. Also see [“Constraints when using read-only variables” on page 283](#).

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. You can use data set name format to 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. For a description of the indexing function for virtual tape system parameters, see [“Virtual tape system parameter” on page 288](#).

Type: Alphanumeric

Max value: 142 characters

**&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 that is 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

**&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 288](#) 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



**FS**

VSAM linear zFS data set

**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

**&SECLABL**

Specifies the default security label in the RACF profile of the user or data set if the SECLABEL class is active. Otherwise, the read-only variable will contain a null value.

Type: Literal. The first character must be alphabetic, \$, #, or @.

Max value: 1 - 8 alphanumeric or national (\$, #, @) characters.

**&SECOND\_QTY**

Specifies a secondary allocation quantity from JCL or AMS control cards. It has meaning only when the &SPACE\_TYPE variable is defined. In conjunction with &SPACE\_TYPE, &SECOND\_QTY allows the DATACLAS ACS routine to make the appropriate DATACLAS assignment. You must modify your existing ACS routines if you want to use these two variables.

**&SIZE**

The primary amount of space (in KB or MB) requested for a new data set or the amount of space used in an existing data set on a DASD volume (see "Constants" on page 267). See "Using read-only variables" on page 281 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

**&SPACE\_TYPE**

Specifies the allocation unit from JCL or AMS control cards to be used with the secondary space allocation amount. With the &SECOND\_QTY variable, &SPACE\_TYPE allows the DATACLAS ACS routine to make the appropriate DATACLAS assignment. You must modify your existing ACS routines if you want to use these two variables.

Valid values: TRK, CYL, K, M, U, BLK, and blank

**&STEP**

The step name of the job running at the time ACS routines was invoked, if available. First character of step name is restricted to alphabetic and special characters, no numeric characters, similar to the JCL step name restriction.

Type: Literal

Max Value: 8 characters

Using &STEP within a comparison has additional checks to be considered valid. When &STEP is compared within a statement, the literal string first character only accepts alphabetic and special characters, no numeric characters. This restriction reflects the JCL job step name restriction.

For example,

IF &STEP = '123STEP' THEN... is invalid

IF &STEP = 'STEP123' THEN... is valid

#### **&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 281](#) 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 281](#) for usage information.

Type: Literal

Max value: 8 characters

#### **&TIME**

The current local time before the ACS routines were invoked. Formatted as a literal string 'HH:MM:SS' where:

- HH = Hours

- MM = Minutes

**Note:** When comparing &TIME with literal string within the ACS routines, ensure proper formatting with colon symbol.

Type: Literals

Max Value: '23:59'

Using &TIME within a comparison has additional checks to be considered valid. When &TIME is compared in a statement, the literal string must match the format of 'HH:MM'. The literal string being compared also must be within range of the standard time format for hours (00-23), and minutes (00-59).

For example,

IF &TIME < '24:30' THEN... is invalid

IF &TIME > '1230' THEN... is invalid

IF &TIME > '12:30' THEN ... is valid

IF &TIME > '00:30' THEN... is valid

#### **&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 162](#).

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 recovers, or the user ID of the DFSMSHsm address space. If the environment is recall or recover, DFSMSHsm sets &USER for both authorized and non-authorized users if it is available. Therefore, do not rely on a null value for &USER to determine if a user is a DFSMSHsm authorized user or not. (Refer to [“Determining Distributed FileManager/MVS data set creation requests” on page 288](#) for Distributed FileManager/MVS usage information.)

Type: Literal

Max value: 8 characters

### **&USER\_ACSVAR**

Is a user-defined variable. The values for &USER\_ACSVAR are positional and are derived from the USER\_ACSVAR parameter, which is defined in the IGDSMSxx member of PARMLIB. They can be altered by the SETSMS USER\_ACSVAR command.

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.

As input to data class ACS routine, &SIZE and &MAXSIZE are calculated from space information from JCL or IDCAMS, as follows:

- For non-VSAM data sets:
  - $\&SIZE = P + \text{Directory (if any)}$ .
  - $\&MAXSIZE = P + (S*15) + \text{Directory (if any)}$ .
- For VSAM data sets:
  - $\&SIZE = P$ .
  - $\&MAXSIZE = P + (S*254)$ .

After data class is derived by the data class ACS routine,  $\&SIZE$  and  $\&MAXSIZE$  are recalculated depending on whether Extent Constraint Removal is specified or not in the selected data class before calling subsequent data class routines, as follows:

- For non-VSAM data sets, if no space is specified on JCL, then the space values defined in the selected data class are used to recalculate  $\&SIZE$  and  $\&MAXSIZE$  as follows:
  - $\&SIZE = P + \text{Directory (if any)}$ .
  - $\&MAXSIZE = P + (S*15) + \text{Directory (if any)}$ .
- For VSAM data sets, regardless whether space is specified or not,  $\&MAXSIZE$  is recalculated depending on the specification of Extent Constraint Removal attribute as follows:
  - If Extent Constraint Removal = NO, then  $\&MAXSIZE = P + (S*122)*\text{volcnt}$ .
  - If Extent Constraint Removal = YES, and the ADD' '1 Volume Amount in Data Class = P, then  $\&MAXSIZE = ((P+S*122))*\text{volcnt}$ .
  - If Extent Constraint Removal = YES, and the ADD' '1 Volume Amount in Data Class = S, then  $\&MAXSIZE = ((P+S*122)) + ((S*123)*(\text{volcnt}-1))$ .

## Using OAM read-only variables

$\&MEMN$ ,  $\&MEMHLQ$ ,  $\&MEMLLQ$  and  $\&MEMNQUAL$  are used to name an object in a collection.

**Restriction:** 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.

$\&DB2SSID$  provides the SSID of the Db2 subsystem that is associated with the instance of OAM that is invoking the ACS routine.

**Restriction:**  $\&DB2SSID$  is only valid in the STORE, CHANGE, and CTRANS environments in the Storage Class and Management Class ACS routines. It is also only applicable to the STORE environment in the Storage Group ACS routine, 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 groups 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

During rename processing, the STORCLAS ACS routine is not called or redriven. However, redriving the MGMTCLAS ACS routine on a data set rename invokes the management class ACS routine when an SMS-managed cluster, generation data set (GDS) or non-VSAM data set is renamed. Catalog management invokes MGMTCLAS ACS routines during rename processing. Refer to [“Access Method Services” on page 169](#) for more information.

## 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:

1. 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

2. 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  
&REORG  
&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 282.](#)

## 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 290](#) for details.

## Comparison rules

The following rules apply to comparisons:

- 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.
- Numerics are right justified.
- Literals are left justified and padded with blanks.
- 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.
- 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 270](#) for maximum lengths.
- Some read-only variables have additional rules applied to be considered valid such as format checks and value range checks. See description for individual checks under [“Read-only variables” on page 270](#).

## 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 indexing functions, which allow you to make class selections based on specific details about each data set.

*Table 25. Indexing Functions for Read-Only Variables*

<b>Selection Routine</b>	<b>Data Set Qualifier</b>	<b>&amp;ALLVOL &amp;ANYVOL</b>	<b>Accounting Information</b>	<b>&amp;USER_ACSVAR</b>
<b>Data class</b>	Yes	Yes	Yes	Yes
<b>Storage class</b>	Yes	Yes	Yes	Yes
<b>Management class</b>	Yes	Yes	Yes	Yes
<b>Storage group</b>	Yes	Yes	No	Yes

## 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 numbers 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 48 on page 287](#):

```
PROC STORCLAS
  IF &ALLVOL = IMS* THEN
    (code left out)
  SELECT
    WHEN (&ANYVOL = TSO*)
      (code left out)
  END
END
```

*Figure 48. 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.

For a detailed explanation of the IF statement, refer to [“IF statement” on page 293](#).

## **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 that is separated by commas in the account information. To request indexing, specify the variable with (n), where n is the field number (1 through 71). &ACCT\_JOB or &ACCT\_STEP without an index accesses the first field of the accounting information by default.

The following are examples of using &ACCT\_JOB and &ACCT\_STEP.

First field of the JOB accounting information:

```
&ACCT_JOB(1)
```

First field of the JOB accounting information (default):

```
&ACCT_JOB
```

Third field of the JOB accounting information:

```
&ACCT_JOB(3)
```

First field of the STEP accounting information (default):

```
&ACCT_STEP
```

Fifth field of the STEP accounting information:

```
&ACCT_STEP(5)
```

## &USER\_ACSVAR indexing function

Allows you to reference a specific positional value in the USER\_ACSVAR parameter in PARMLIB member IGDSMSxx. Positional values in the USER\_ACSVAR parameter are separated by commas. To request indexing, specify &USER\_ACSVAR(*n*), where *n* is the position (a number from 1 through 3). &USER\_ACSVAR without an index accesses the first positional value of the USER\_ACSVAR parameter. An index of other than 1 through 3 generates an error condition with a reason code 2032 when you run the ACS routine.

## Virtual tape system parameter

Allows you to access the subfields in the virtual tape system parameter set by the Pre-ACS Installation Exit. The parameter is defined as a 256-byte character string including one or more sets of 2-byte length field followed by the value. The use of &MSPARM(*n*), where *n* is the field number, indicates that indexing is requested. If &MSPARM is specified without indexing, then the default is to access the first field of the parameter (for example, &MSPARM1) result by default).

&MSPARM	first field of the parameter (default)
&MSPARM(1)	first field of the parameter
&MSPARM(4)	fourth field of the parameter

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 49 on page 288](#) 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 49. 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 50 on page 289](#) 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 50. 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 51 on page 289 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 51. 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)>*]

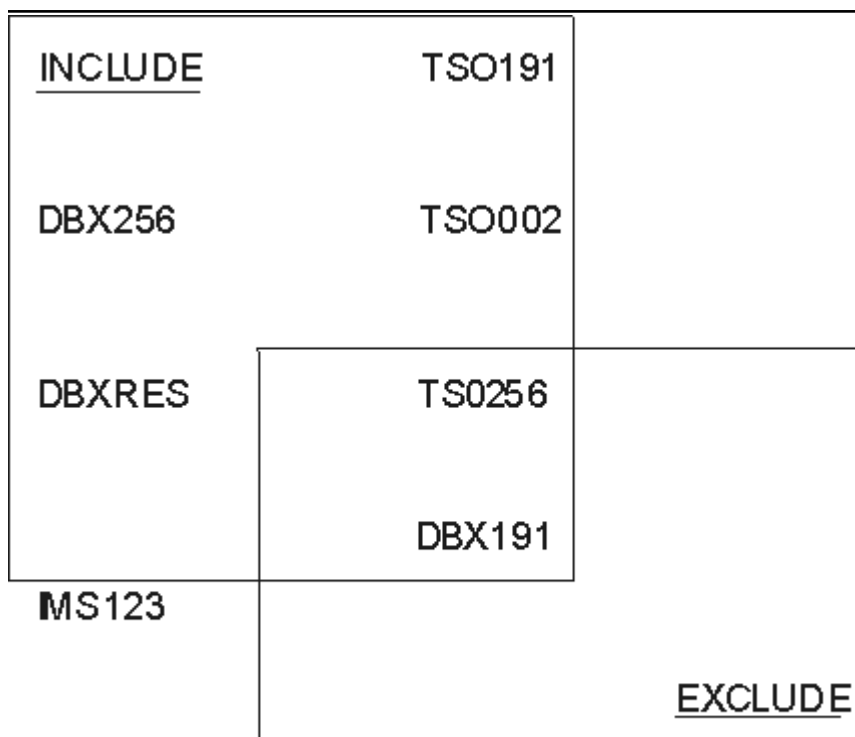
- *name* is mandatory and can be up to 31 alphanumeric characters in length. You can also use an underscore, \_, but it cannot be the first character. In the FILTLIST, you can optionally precede the name with an ampersand, & When referring to the FILTLIST in the body of the routine, you must always precede 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 52 on page 291 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 52. Example of a FILTLIST Statement



DA6S2016

Figure 53. Using the INCLUDE and EXCLUDE

In the environment shown in Figure 53 on page 291, 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

## 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 26 on page 292](#) summarizes the possible assignments of *value*.

Table 26. Read-Write Variable Assignments

	One Name in Single Quotation Marks	Null Value	RACF Read-Only Variable Name	List of Quoted Names Separated by Commas
<b>Storage group</b>	X			X
<b>Management class</b>	X	X	X	
<b>Storage class</b>	X	X	X	
<b>Data class</b>	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 54 on page 293 shows an example of a DO statement:

```
IF &HLQ='PAYROLL' THEN
DO
  WRITE 'No Payroll allowed'
  EXIT CODE(1)
END
```

Figure 54. 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.

For information on what constitutes a valid relational expression, refer to [“Comparison rules” on page 285](#).

Figure 55 on page 293 shows two examples of IF statements:

Example of a specified ELSE:	Example of a null ELSE:
<pre>IF &amp;DSOWNER = 'BILL' THEN   SET &amp;STORCLAS = 'ONE' ELSE   SET &amp;STORCLAS = 'TWO'</pre>	<pre>IF &amp;DSOWNER = 'BILL' THEN   SET &amp;STORCLAS = 'ONE' ELSE   DO END   SET &amp;STORCLAS = 'TWO'</pre>

Figure 55. 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 56 on page 295](#) 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 56. 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 57 on page 295 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 57. Example of an EXIT Statement

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.

When using TSO ALLOCATE or IDCAMS ALLOCATE command, the text of the WRITE message displays only if the allocation fails.

**Note:** In ISPF 3.2, the WRITE message displays on allocation success or failure.

**Exception:** The WRITE message is never displayed for any product or application that uses Dynamic Allocation (for example: DFSMSdss, DFSORT SORTWKnn data sets). However, if the parameter ACS\_ROUTINE\_DEBUG in SYS1.PARMLIB(IGDSMSxx) is enabled, all WRITE messages may be displayed.

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.

ACS\_ROUTINE\_DEBUG is a feature that will allow a user to externalize all ACS WRITE statements to the job log. The feature can be controlled by disabling the feature, enabling the feature for all jobs, or enabling the feature for a specific jobname. These statements will appear as additional messages outside of the original message chain, which has a 5-message maximum per routine. This improves traceability while debugging the ACS routines. Descriptions of the SMS command and PARMLIB member can be found in *z/OS MVS System Commands* and *z/OS MVS Initialization and Tuning Reference* respectively.

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 or a literal onto a subsequent line. The closing single quotation mark signifies the end of text. A variable can be referenced on a subsequent line after the closing quotation mark without a continuation character.

For example, the following WRITE message:

```
WRITE 'This line's short.'
```

displays as:

```
This line's short.
```

The following WRITE message:

```
WRITE 'THE DATACLAS IS EQUAL TO'  
      &DATACLAS
```

displays as:

```
THE DATACLAS IS EQUAL TO DEFAULT
```

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
```

If only one storage group is assigned to &STORGRP, the WRITE statement will print that single construct name. When multiple storage group constructs are assigned to &STORGRP, the first construct will be printed, followed by additional lines printing the subsequent constructs. For example, if the values for &STORGRP are 'SG1', 'SG2', and 'SG3', the following WRITE statement:

```
WRITE ' &STORGRP IS '&STORGRP' '
```

displays:

```
IGD01010I &STORGRP IS SG1  
REST OF STORGRPS: SG2, SG3
```

Example ACS code assigning one storage group:

```
SET &STORGRP = 'S1P01'WRITE 'SET SG IS ... '&STORGRP' '
```

ACS WRITE will print:

```
'SET SG IS ... S1P01'
```

Example ACS code assigning two storage groups:

```
SET &STORGRP = 'S1P01','S1P02'  
WRITE 'SET SG IS ... '&STORGRP' '
```

ACS WRITE will print:

```
'SET SG IS ... S1P01'  
'REST OF STORGRPS: S1P02'
```

Example ACS code assigning 15 storage groups:

```
SET &STORGRP = 'S1P01','S1P02','S1P03','S1P04','S1P05','S1P06','S1P07','S1P08',  
'S1P09','S1P10','S1P11','S1P12','S1P13','S1P14','S1P15'  
WRITE 'SET SG IS ... '&STORGRP' '
```

ACS WRITE will print:

```
'SET SG IS ... S1P01'  
'REST OF STORGRPS: S1P02, S1P03, S1P04, S1P05, S1P06, S1P07'  
'S1P08,S1P09, S1P10, S1P11, S1P12, S1P13, S1P14, S1P15'
```

## END statement

The END statement concludes an ACS routine, a DO group, or a SELECT statement. [Figure 58 on page 297](#) shows an example of an END statement:

```
PROC STORCLAS  
  (source code)  
END
```

*Figure 58. Example of an END Statement*

## LOGWRITES statement

Use the LOGWRITES statement to facilitate the debugging and monitoring the execution of ACS routines. If an execution of an ACS routine executes a LOGWRITES command, it causes a LOGREC entry to be created which includes the JOBNAME, STEPNAME (only on z/OS 2.4 and above), USERID, and any WRITE statements that are executed.

The LOGREC entry is only created if ACS\_LOCWRITES=ENABLE is specified in an IGDSMSXX statement or SETSMS command. LOGWRITES command must not be included in your ACS routine until this support is installed on all systems in your SMS PLEX.

For example,;

```
PROC 1 DATACLAS
WRITE ' *** &DSN IS '&DSN' *** WITH ACS LOG'
IF &DATACLAS = 'TESTING' THEN
  DO
    LOGWRITES
    WRITE 'ACSLOG INVOKED FOR '&DATACLAS
  END
SET &DATACLAS='MYDC0001'
END
```

## Sample ACS routine

Figure 59 on page 298 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 59. Production ACS Routine for Storage Class

The FILTLIST VALID\_UNITS INCLUDE statement in Figure 59 on page 298 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 59 on page 298 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.

---

## Chapter 20. 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

**CP**

Copy Pool

**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 DFSMSdss does not display an entry panel.

Table 27 on page 300 and Table 28 on page 303 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 27. ISMF Commands

Command	Minimum Abbreviation	Description	Application	Source
<b>ACTIVATE</b>	AC	Copy the contents of an SCDS into an ACDS and activate it, or activate an existing ACDS.	CDS	DFSMSdfp
<b>ALTER</b>	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
<b>AUDIT</b>	AU	Verify the location of volumes in 3995 optical libraries or automated tape libraries.	OVOL, TVOL, TL	DFSMSdfp
<b>BOTTOM</b>	BOT	Scroll to the bottom of the entries.	All except ACS and CDS	DFSMSdfp
<b>CANCEL</b>	CA	Return to the previous dialog without performing any of the current dialog functions.	DC, SC, MC, SG, ACS, CDS, AG, LB, DV, LA, CP	DFSMSdfp
<b>CLEAR</b>	CLE	Reset line operator history.	All except ACS and CDS	DFSMSdfp
<b>CLEAR ALL</b>	CL ALL	Clear all pages on selection entry panels and filter panels.	DS, DVOL	DFSMSdfp
<b>CLEAR PAGE</b>	CL PA	Clear the current page on selection entry panels and filter panels.	DS, DVOL, OVOL, LA, TVOL	DFSMSdfp
<b>CLEAR PAGE<sub>x</sub></b>	CL PAGE <sub>x</sub>	Clear a page on selection entry panels and filter panels where x is the page number.	DS, DVOL	DFSMSdfp
<b>COMPRESS</b>	COM	Reclaim embedded unused space from a list of PDSs.	DS	DFSMSdss

Table 27. ISMF Commands (continued)

Command	Minimum Abbreviation	Description	Application	Source
<b>COPY</b>	COP	Copy a list of data sets to a DASD volume of like or unlike device type.	DS	DFSMSDss
<b>DOWN n</b>	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
<b>DSUTIL</b>	DSUTIL	Invoke PDF data set utility functions.	DS	DFSMSdfp
<b>DUMP</b>	DU	Dump data sets to tape, DASD, or mass storage volumes.	DS	DFSMSDss
<b>END</b>	END	Exit the current ISMF function or panel and return to the previous panel.	All	DFSMSdfp
<b>ERTB</b>	ER	Display the ISMF Error Table.	All	DFSMSdfp
<b>FILTER</b>	FIL	Tailor the list to include only specific entries.	DS, DVOL, LA	DFSMSdfp
<b>FILTER CLEAR</b>	FIL C	Clear the filter entries but bypass the entry panel.	DS, DVOL, LA	DFSMSdfp
<b>FIND</b>	FIN	Find a specific data column.	All except ACS and CDS	DFSMSdfp
<b>FOLD</b>	FO	Extend the data set name data column.	DS	DFSMSdfp
<b>HELP</b>	HELP	Request the help panels associated with the current panel.	All	DFSMSdfp
<b>LEFT</b>	L	Scroll left the specified number of columns.	All except ACS and CDS	DFSMSdfp
<b>LIBRARY</b>	LIBRARY	Invoke PDF library utility.	DS	DFSMSdfp
<b>LISTPRT</b>	LISTP	Prints generated or saved ISMF lists	All that can generate a list	DFSMSdfp

Table 27. ISMF Commands (continued)

Command	Minimum Abbreviation	Description	Application	Source
<b>PROFILE</b>	P	Invoke the ISMF profile.	All except the ISMF Profile Application, the ISMF Menu panel or an abend panel	DFSMSdfp
<b>QSAVE</b>	QS	Save query criteria	DS, DVOL	NaviQuest
<b>QRETRIEV</b>	QR	Retrieve query criteria	DS, DVOL	NaviQuest
<b>REFRESH</b>	REF	Display the updated list.	All except ACS and CDS	DFSMSdfp
<b>RELEASE</b>	REL	Free unused space at the end of each of the data sets in a list.	DS	DFSMSdss
<b>RESHOW</b>	RESH	Redisplay all list entries removed by the HIDE line operator.	All except ACS and CDS	DFSMSdfp
<b>RESTORE</b>	REST	Restore data sets that have been dumped by DFSMSdss.	DS	DFSMSdss
<b>RETURN</b>	RETURN	Return to the primary ISMF option menu.	All	DFSMSdfp
<b>RIGHT</b>	RI	Scroll right the specified number of columns.	All except ACS and CDS	DFSMSdfp
<b>SAVE</b>	SA	Save a copy of the current list in the ISPF output table.	All except ACS and CDS.	DFSMSdfp
<b>SORT</b>	SO	Organize lists based on entries in specific data columns.	All except ACS and CDS	DFSMSdfp
<b>TOP</b>	TOP	Scroll to the top of the entries.	All except ACS and CDS	DFSMSdfp
<b>TSO Commands and CLISTs</b>		Invoke TSO commands and CLISTs.	All	DFSMSdfp
<b>UP</b>	U	Scroll backward the specified number of entries. UP MAX scrolls to the top of the entries.	All except ACS and CDS	DFSMSdfp



Table 27. ISMF Commands (continued)

Command	Minimum Abbreviation	Description	Application	Source
<b>VALIDATE</b>	V	Check the completeness and consistency of an entire SCDS.	CDS	DFSMSdfp
<b>VIEW</b>	VI	Choose the display order of the columns on list panels.	All except ACS and CDS	DFSMSdfp

Table 28. ISMF Line Operators

Line Operator	Minimum Abbreviation	Description	Application	Source
<b>ABACKUP</b>	AB	Specify the parameters for an ABACKUP command and issue the command to DFSMSHsm.	AG	DFSMSHsm
<b>ALTER</b>	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, CP	DFSMSdfp
<b>ANALYZE</b>	ANAL	Examine the device or data on a volume to determine if any errors exist.	DVOL	ICKDSF
<b>AUDIT</b>	AU	Verify the location of optical volumes in IBM 3995 optical and automated tape libraries.	OVOL, LB, TVOL, TL	DFSMSdfp
<b>BROWSE</b>	B	View a sequential data set or a member of a PDS.	DS	DFSMSdfp

Table 28. ISMF Line Operators (continued)

Line Operator	Minimum Abbreviation	Description	Application	Source
<b>BUILDIX</b>	BUI	Change a volume from MVS format VTOC(OSVTOC) to an indexed format VTOC(IXVTOC) or vice versa.	DVOL	ICKDSF
<b>CATLIST</b>	CATLIST	Invoke IDAMS LISTCAT and browse the output.	DS	DFSMSdfp
<b>CGCREATE</b>	CGCREATE	Allow I/O activity to resume on the volumes residing in the logical subsystems receiving the command.	DVOL	DFSMSdss
<b>CLIST</b>	CLI	Call a TSO CLIST.	DS, DVOL, OVOL, TVOL, TL	DFSMSdfp
<b>COMPRESS</b>	COM	Reclaim embedded unused space from a PDS.	DS, DVOL	DFSMSdss
<b>CONDENSE</b>	CON	Free unused space at the end of a data set; compress a PDS.	DS	DFSMSdsm
<b>CONSOLID</b>	CONS	Copy data from one DASD volume to another	DVOL	DFSMSdss
<b>CONTROL</b>	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
<b>CONVERTV</b>	CONV	Convert DASD volumes into SMS or out of SMS.	DVOL	DFSMSdss
<b>COPY</b>	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, CP	DFSMSdfp or DFSMSdss
<b>DEFRAG</b>	DEFR	Reduce free-space fragmentation on a DASD. volume	DVOL	DFSMSdss

Table 28. ISMF Line Operators (continued)

Line Operator	Minimum Abbreviation	Description	Application	Source
<b>DELETE</b>	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, CP	DFSMSdftp or DFSMSHsm
<b>DISPLAY</b>	DI	Display an SMS class and its attributes.	DC, SC, MC, AG, LB, DV, TL, CP	DFSMSdftp
<b>DUMP</b>	DU	Dump a data set or volume to tape or DASD.	DS, DVOL	DFSMSdss
<b>EDIT</b>	E	Edit a sequential data set or member of a PDS.	DS	DFSMSdftp
<b>EJECT</b>	EJ	Eject an optical or tape volume from a library.	OVOL, TVOL, TL	DFSMSdftp
<b>ERASE</b>	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, CP	DFSMSdftp or DFSMSHsm
<b>HALTERDS</b>	HA	Change the number of backup versions of a data set; change frequency of backup.	DS	DFSMSHsm
<b>HBACKDS</b>	HBA	Create a backup version of a data set.	DS	DFSMSHsm
<b>HBDELETE</b>	HBD	Delete backup versions of a data set.	DS	DFSMSHsm
<b>HDELETE</b>	HDE	Delete a migrated data set.	DS	DFSMSHsm
<b>HIDE</b>	HI	Remove a list entry from display.	All except ACS and CDS.	DFSMSdftp
<b>HMIGRATE</b>	HM	Migrate a data set to DFSMSHsm level one or level two volume.	DS	DFSMSHsm
<b>HRECALL</b>	HRECA	Recall a data set that has been migrated by DFSMSHsm.	DS	DFSMSHsm
<b>HRECOVER</b>	HRECO	Recover a backup version of a data set.	DS	DFSMSHsm
<b>INIT</b>	INI	Initialize a volume.	DVOL	ICKDSF

Table 28. ISMF Line Operators (continued)

Line Operator	Minimum Abbreviation	Description	Application	Source
<b>INSPECT</b>	INS	Detect defects in volume track surface.	DVOL	ICKDSF
<b>INSTALL</b>	INST	Install an HDA replacement and physical movement of IBM DASD.	DVOL	ICKDSF
<b>LIST</b>	LI	Retrieve a list that was saved with the SAVE command.	LA	DFSMSdfp
<b>LISTSYS</b>	LISTS	List the systems associated with a given storage group.	SG	DFSMSdfp
<b>LISTVOL</b>	LISTV	List the volumes associated with a given storage group, optical library or tape library.	SG, LB, TL	DFSMSdfp
<b>MESSAGE</b>	MES	Display message text for the last operation performed on a list entry.	All	DFSMSdfp
<b>PPRCOPY</b>	PPRC	Allows synchronous copying of a DASD volume from one subsystem to another subsystem volume.	DVOL	ICKDSF
<b>RAUTH</b>	RAUTH	Provide remote access codes.	DVOL	DFSMSdfp
<b>RECOVER</b>	REC	Recover a backup copy of all the objects on a primary or backup optical cartridge.	OVOL	DFSMSdfp
<b>REFORMAT</b>	REF	Change the volume serial number or owner ID of a volume.	DVOL	ICKDSF
<b>RELEASE</b>	REL	Free unused space at the end of data sets.	DS, DVOL	DFSMSdss
<b>REMAP</b>	REM	Reconstruct inventory in a real IBM 3995 optical library.	LB	DFSMSdfp
<b>REPEAT</b>	=	Repeat the last operator command that was implemented.	All except ACS and CDS.	DFSMSdss DFSMSdfp

Table 28. ISMF Line Operators (continued)

Line Operator	Minimum Abbreviation	Description	Application	Source
<b>RESTORE</b>	REST	Restore data sets that have been dumped by DFSMSdss.	DS, DVOL	DFSMSdss
<b>REVAL</b>	REV	Perform track validation of medial initialization.	DVOL	ICKDSF
<b>SECURITY</b>	SE	Invoke a RACF panel to protect data sets, storage classes, or management classes.	DS, SC, MC, LB, DV, TL	RACF
<b>SETCACHE</b>	SETC	Manage storage control characteristics.	DVOL	DFSMSdfp
<b>STATUS</b>	ST	Allows the display of up to 32 SMS and MVS volume statuses.	DVOL	DFSMSdfp
<b>TRKFMT</b>	TRKF	Performs track-related functions.	DVOL	ICKDSF
<b>TSO Commands and CLISTs</b>		Invoke TSO commands and CLISTs.	All	DFSMSdfp
<b>VTOCLIST</b>	VTOCLIST	Invoke IEHLIST LISTVTOC.	DS	DFSMSdfp



---

# Chapter 21. SETCACHE functions and device information

This topic 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 a nonindexed 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 \(ICKDSF\) 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
<b>Manage storage control unit characteristics</b>	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.

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.

**Restriction:** 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 redisplayes 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
<b>Provide remote access codes</b>	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.

---

## Chapter 22. 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 the DFSMSrmm Report Generator for creating customized reports and 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.

Perform the following steps to generate the JCL for the access method services DCOLLECT command.

1. Select option C, Data Collection, on the ISMF Primary Option Menu for storage administrators. This displays the Data Collection Entry panel.  
  
-----
2. Supply values on the 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 for Select Data Collection options.  
  
-----
3. 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.  
  
-----
4. Supply values for one or more volumes or storage groups about which DCOLLECT is to collect information..  
  
-----
5. If you want to specify volumes to exclude, use DOWN command to reach the last page of the Data Collection Entry panel.  
  
-----
6. Supply values to specify volumes to exclude.  
  
-----
7. 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.

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.

On the second page, you 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.



# Chapter 23. 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 or the volume level, from all the other data sets in the same group.

## Overview

To use 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 attempts 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 specifies whether separation is at the PCU or volume level, whether it is required or preferred, and includes a list of data set names to be separated from each other during allocation.

**Restriction:** You cannot use data set separation when allocating non-SMS-managed data sets or with any process that does not use SMS volume selection, such as a full-volume copy utility (for example PPRC).

**Recommendation:** Use data set separation only for a small set of mission-critical data for the following reasons:

- When separation is at the PCU level, 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.
- The use of wildcard characters in the data set names that need to be separated from each other might drastically reduce the number of eligible volumes or result in allocation failure 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 322](#).

For information about specifying the name of the data set separation profile to the base configuration, see [“Specifying the DS separation profile” on page 21](#).

The following table lists the topics that are contained in this section:

Topic
<a href="#">“Syntax for creating a data set separation profile” on page 317</a>
<a href="#">“Data set requirements for a data set separation profile” on page 320</a>
<a href="#">“Creating multiple data set separation profiles” on page 320</a>
<a href="#">“Using data set separation with generation data groups and striping” on page 321</a>
<a href="#">“Environmental conditions affecting data set separation” on page 321</a>
<a href="#">“Factors affecting code path length” on page 322</a>

## Syntax for creating a data set separation profile

A data set separation profile contains one or more data set separation groups. Use the following syntax to create a data set separation group:

```
SEPARATIONGROUP|SEP (PCU |{VOLUME|VOL})  
TYPE ({REQUIRED|REQ|R} | {PREFERRED|PREF|P})
```

```
DSNLIST|DSNS|DSN (data-set-name,data-set-name[,data-set-name,...]  
)
```

where

#### **SEPARATIONGROUP(PCU)**

Indicates that separation is on the PCU level.

#### **SEPARATIONGROUP(VOLUME)**

Indicates that separation is on the volume level. VOLUME may be abbreviated as VOL.

#### **TYPE(REQUIRED)**

Indicates that separation is required. SMS fails the allocation if the specified data set or data sets cannot be separated from other data sets on the specified level (PCU or volume). REQUIRED may be abbreviated as REQ or R.

#### **TYPE(PREFERRED)**

Indicates that separation is preferred. SMS allows the allocation if the specified data set or data sets cannot be separated from other data sets on the specified level. SMS allocates the data sets and issues an allocation message that indicates that separation was not honored for a successful allocation. PREFERRED may be abbreviated as PREF or P.

#### **DSNLIST(data-set-name,data-set-name[,data-set-name,...])**

Specifies the names of the data sets that are to be separated. You must specify at least 2 names. The data set names 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. Wildcard characters are supported, beginning with the third qualifier. For example, you could specify BJONES.TEST.\* but not BJONES.\*. The wildcard characters are as follows:

**\***

Indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.

**\*\***

Indicates that zero or more qualifiers can occupy that position. When \*\* is the last qualifier, one or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a period or a blank.

**%**

Indicates that exactly one alphanumeric or national character can occupy that position. You can specify up to eight % characters in each qualifier.

If only one data set name is specified with DSNLIST, the data set name must contain at least one wildcard character.

## **Deprecated syntax**

The following earlier form of the syntax for SEPARATIONGROUP is tolerated by z/OS V1R11. It supports separation at the PCU level only.

```
SEPARATIONGROUP|SEP  
FAILLEVEL|FAIL ({PCU|NONE})  
DSNLIST|DSNS|DSN (data-set-name[,data-set-name,...])
```

You specify either required or preferred data set separation by using one of the values for the FAILLEVEL keyword:

#### **FAILLEVEL(PCU)**

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.

For the current syntax, see TYPE(REQUIRED).



**FAILLEVEL(NONE)**

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.

For the current syntax, see TYPE(PREFERRED).

When this syntax is used, data set names listed in a data set separation group must not contain quotation marks or wildcards, 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.

**Migrating to the new syntax**

If you have the deprecated syntax in your data set separation profile, consider updating the profile to use the current syntax. Toleration PTFs are available that provide support for the z/OS V1R11 syntax for PCU-level separation on lower level systems, with the exception of wildcard characters in data set names. With the PTFs, separation groups defined with syntax that is not supported, such as SEP(VOL) or wildcard characters in the data set name list, are ignored.

**Example**

The following examples show definitions that are equivalent with the deprecated and current syntax:

Table 29. PCU-Level Syntax Examples	
Deprecated Syntax	Current Syntax
SEP - FAIL(PCU) - DSNLIST(A.B.C1, A.B.C2);	SEP(PCU) - TYPE(REQ) - DSNLIST(A.B.C1, A.B.C2);

**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(PCU) -  
/*SMS Control Data Sets*/-
```

;

You must use this termination character to end a SEPARATIONGROUP statement. For example:

```
SEP(PCU) -  
TYPE(REQ) -  
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(PCU) -  
TYPE(PREF) -  
DSNLIST(SMS.PROD.SCDS, SMS.PROD.ACDS, SMS.PROD.COMMDS);  
  
/*-----  
JES CHECKPOINT Data Sets  
-----*/  
SEP(PCU)  
TYPE(REQ)  
DSNS(SYS1.JESCKPT1, - !primary  
      SYS1.JESCKPT2, - !secondary  
      SYS1.JESCKPT3); !tertiary
```

## Volume selection for data set separation by volume

When separation by volume is requested for a data set, the order of preference for volume selection is as follows:

- Candidate volumes that are in different extent pools than volumes on the separation volume list are most preferred
- Candidate volumes that are not on the separation volume list but are in the same extent pools as the volumes on the separation volume list are the next preferred
- When separation by volume is required, SMS must allocate the data set onto the volumes that are not used by any of the data sets in the same separation group
- When separation by volume is preferred, the candidate volumes that are already occupied by other data sets in the separation group are ranked lower and are less preferred.

## Data set requirements for a data set separation profile

The profile data set has these requirements:

- Organization can be either sequential or partitioned
- Record format must be fixed or fixed-block (RECFM=FB).
- It must be cataloged. Catalog the profile data set in the MASTERCAT to avoid a deadlock between SMS and CATALOG during IPL with contention in the SMS ASID, or a SYSZTIOT might occur.

When modifying the profile, use a disposition of OLD (DISP=OLD). The data in the profile might be corrupted if you use a disposition SHR.

## 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 on the PCU level 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 on the PCU level, 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 non-migrated GDG generations.

For more information about GDGs, refer to the discussion of GDG Management Attributes in “[Defining management class migration attributes](#)” on page 73. For more information about data set striping, refer to “[Striping volume selection](#)” on page 105.

## Environmental conditions affecting data set separation

---

Under the following environmental conditions, SMS might not separate 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 DFSMSHsm 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 DFSMSHsm migration to level 1 or level 2 storage.
- SMS *does* perform data set separation during DFSMSHsm 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 use of wildcard characters in the separation data set names
- The number of volumes that are eligible for selection.

---

## Chapter 24. Using NaviQuest

DFSMS NaviQuest is a data and storage management tool for implementing, testing, and verifying the 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 SMS 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 SMS testing functions in batch
- Update SMS configuration values in batch

NaviQuest lets you run jobs in batch to list, alter, delete or define a data class, storage class, management class, storage group, aggregate group, or copy pool. You can also use batch mode to display a data class, storage class, management class, and aggregate group.

- Create reports interactively or in batch

NaviQuest reports can be used during your SMS 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 SMS 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 SMS require that ACS routines assign the correct data class, storage class, management class, and storage group to each data set. Implementing SMS, or any change that affects the SMS environment, must be tested. The results from the SMS 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 SMS 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.

---

### Starting NaviQuest

Start NaviQuest by selecting option 11, Enhanced ACS Management, from the ISMF Primary Option Menu. This displays the NaviQuest Primary Option Menu.

When you select one of the options from the NaviQuest Primary Option Menu, each successive panel guides 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 topic:

**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 SMS data services.

**data subtype**

Groupings of data sets from a data type needing identical SMS 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.

## Protecting NaviQuest

---

See Chapter 17, “Protecting the Storage Management Subsystem,” on page 217 for information that also applies to NaviQuest. This includes:

- “Protecting ISMF functions” on page 217
- “Locating module names for ISMF applications” on page 219
- “Locating module names for ISMF functions” on page 219
- “Protecting modules” on page 226

## 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 328.

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 ISMF-saved lists, you can select the data sets that you want NaviQuest to enter into the testbed library as SMS 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 SMS 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 SMS 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 (volume mount analyzer) data 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 SMS 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 SMS management, each in a separate phase of the implementation process.

Figure 60 on page 326 shows six data types to be managed by SMS.

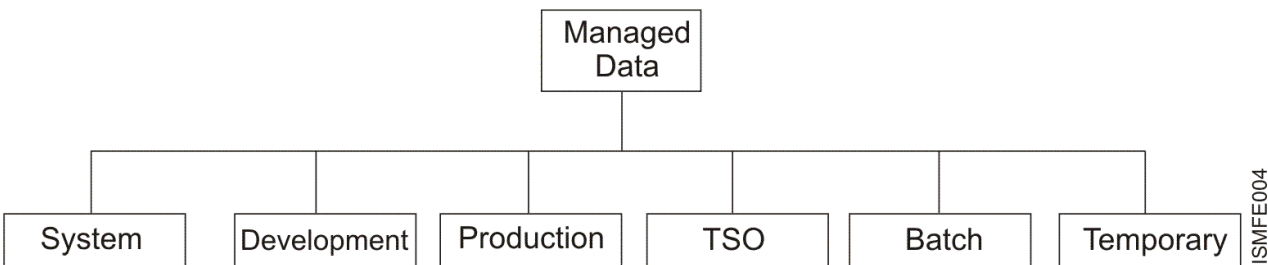


Figure 60. Typical Data Types

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 61 on page 326 is an example of how a TSO data type can be split into different data subtypes.

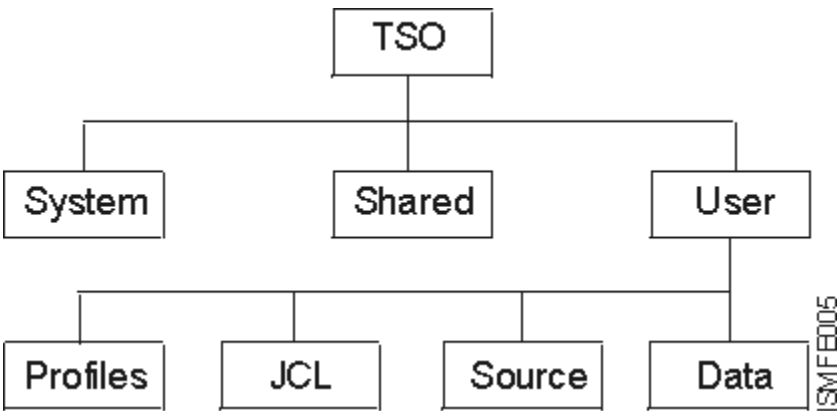


Figure 61. TSO Data Type with Data Subtypes

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 328](#).

- 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 SMS 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” on page 327](#)), 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 325](#) 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 in which 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 327](#) ), you can begin NaviQuest testing. The initial testing establishes the base line test set against data sets that will never be SMS 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 SMS implementation, one subtype at a time. Normally, a SMS 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 SMS 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 SMS management and require the same SMS services.

From the ISMF Primary Option menu, "Data Set" (option 1) offers you two ways to generate data sets samplings:

- From a saved listing
- 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 SAVE command on the command line to save the list into a table. For information on the SAVE command, see [z/OS DFSMS Using the Interactive Storage Management Facility](#).

---

2. Generate the test cases.

Use the Test Case Generation Selection Menu panel to turn the ISMF list, DCOLLECT data, SMS data generated by the storage class ACS exit, and VMA data into standard SMS test cases.

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 to 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.

To generate test cases from DCOLLECT data, select option 2, DCOLLECT Data, and then use Test Case Generation from DCOLLECT Data Entry panel.

Enter the following information:

- 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.

To generate test cases from the ACSTST program, select option 3, SMF Data, and then use Test Case Generation from SMF Data Entry panel.

Enter both the data set name containing the system management facility (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.

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.

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.

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 368.](#)
- For option 11.1.2, use the ACBQBAG1 EXEC. See [“Generate test cases from DCOLLECT data: ACBQBAG1” on page 369.](#)
- For option 11.1.3, use the ACBQBAI1 EXEC. See [“Generate test cases from SMF data” on page 369.](#)
- For option 11.1.4, use the ACBQBAO3 EXEC. See [“Generate test cases from VMA extract data: ACBQBAO3” on page 370.](#)

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 335.](#))

---

### 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 16, “Recovering Storage Management Subsystem information,” on page 213.](#)

You can now update the ACS routines to reflect the new data subtype you want migrated to SMS.

---

### 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 390.](#)

---

### 5. Translate and validate the ACS routines.

You must translate and validate (but not activate) the ACS routines. `refids="vs76475 ts64795">`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 with 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 363.

For more information on translation, see [“Translating ACS routines”](#) on page 151. For more information on validation, see [“Validating ACS routines or an entire SCDS”](#) on page 153.

---

## 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 367.

---

## 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. Because you have not yet stored the expected results of the test cases for this data subtype, these test cases appear as exceptions. Later, in step [“9”](#) on page 333, 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.

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

After running the ACS Test Listings Comparison Entry panel, 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” on page 331](#) 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 331](#), 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 371](#).

---

## 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” on page 331](#) or step [“8” on page 332](#)), you must correct the ACS code before returning to step [“3” on page 330](#) 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) from the NaviQuest Primary Option Menu. 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)

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 331](#).

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.

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” on page 331](#)
- Exception PDS created in step [“7” on page 331](#)
- 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 373](#).

---

#### 10. Test the next data subtype in the current phase.

Continue NaviQuest testing for each data subtype in the current SMS implementation phase. This testing either repeats Steps [“1” on page 328](#) through [“10” on page 333](#) or repeats Steps [“3” on page 330](#) through [“10” on page 333](#), 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 SMS 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 14, “Activating Storage Management Subsystem configurations,”](#) on page 179.

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 SMS management. There are several options for doing this data migration:

- DFSMSdss 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.

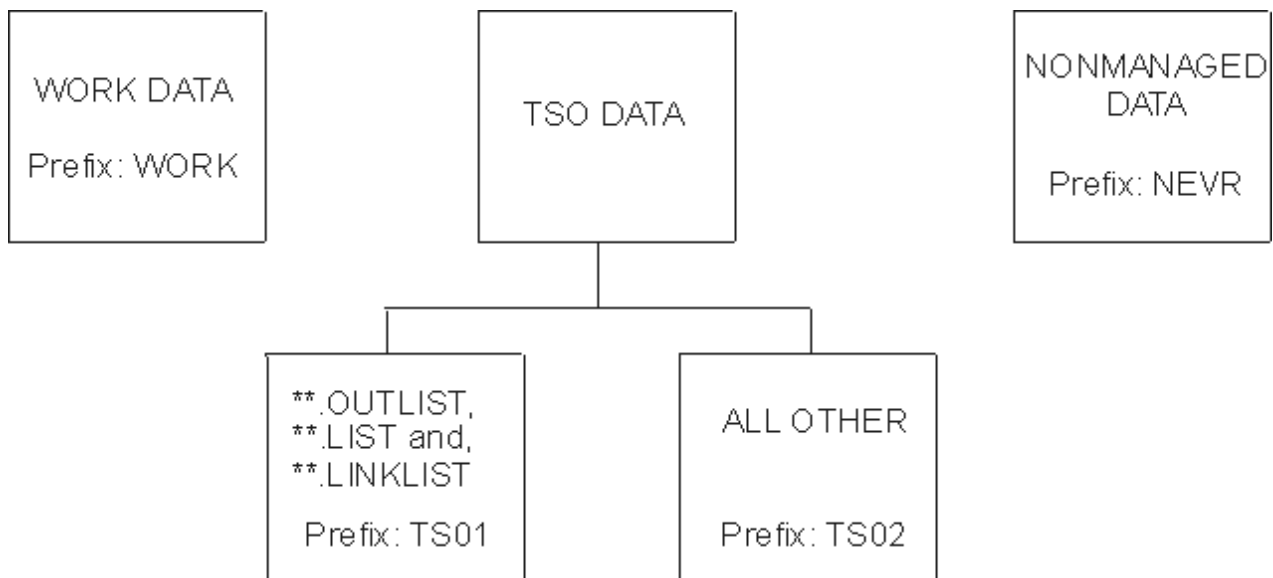
### Data testing

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.

## Activating an ACS routine

---

After you have validated and tested an ACS routine successfully, to make it active and available for systems to invoke, you need to activate it, by activating the SCDS that you translated the ACS routine logic into.

Perform the following steps to activate the validated ACS routine:

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. Select option 3, Validate, on the ACS Application Selection panel. This displays the Validate ACS Routines or Entire SCDS panel.

-----  
2. Specify the name of the SCDS to validate in the CDS NAME field.

3. Select option 5, Activate the SCDS.

4. From the operator console, enter the SETSMS command.

Example: **SETSMS SCDS(dsname)**

## 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 SMS 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.

Perform the following steps to select and run batch jobs.

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 “Using the sample JCL for batch” on page 336, step “2” on page 336, 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.

## Using the 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. The SYS1.SACBCNTL members are listed in [“JCL for ACBJBAOB” on page 340](#).

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 [“JCL for ACBJBAOB” on page 340](#) 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.

**Note:** if you want to clear the optional parameters when performing the ALTER construct function, explicitly specify a blank. For example, specify EXTSGNM( ) in the ACBJBAJ2 job to set the **Extend SG Name** field to blank when the performing the Alter Pool Storage Group function. Or, specify DUMPCLAS( , , , , ) to set all five Dump Classes to empty.

### 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.

[“JCL for ACBJBAOB” on page 340](#) 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 [z/OS ISPF 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.

## **Functions that propagate the return code**

The following functions propagate the return code. Subsequent steps will not execute if COND= is being used.

- Data set lists, reports
- Tape volume lists, reports
- Data class define, alter, display
- Management class define, alter, display
- Storage class define, alter, display
- Storage group define, alter, display
- ACS Object Information display

## **SYS1.SACBCNTL sample JCL library**

Table 30 on page 337 lists the sample job library members:

<i>Table 30. SYS1.SACBCNTL Sample JCL Library Member List</i>	
<b>Member</b>	<b>Function</b>
<b>ACBJBAA1</b>	Aggregate group define/alter/display
<b>ACBJBAB1</b>	Base configuration define/alter/display
<b>ACBJBAC1</b>	ACS test listings comparison report
<b>ACBJBAC2</b>	Translate ACS routines, validate SCDS, test ACS routines, and generate ACS comparison report
<b>ACBJBAC3</b>	Network Connection define / alter / display
<b>ACBJBAC4</b>	Generate Network Connection list and report
<b>ACBJBAC5</b>	Generate Network Connection list and save it
<b>ACBJBAC6</b>	Generate Network Connection report from a saved list
<b>ACBJBAD1</b>	Data class define/alter/display
<b>ACBJBAG1</b>	Generate test cases from previously collected DCOLLECT data ('D' records)
<b>ACBJBAG2</b>	Generate test cases from a previously saved table (data set list)
<b>ACBJBAI1</b>	Generate test cases using data from SMF type 127 records.
<b>ACBJBAI2</b>	Generate data set list and save it in a table
<b>ACBJBAI4</b>	Generate DASD volume list, save it in a table, and save the query
<b>ACBJBAI5</b>	Generate DASD volume list and save it in a table
<b>ACBJBAI7</b>	Generate data set list, save it in a table, and save the query
<b>ACBJBAI8</b>	Generate DASD volume list, save it in a table, and generate a report from it
<b>ACBJBAI9</b>	Generate DASD volume list using a previously saved query and save it in a table

Table 30. SYS1.SACBCNTL Sample JCL Library Member List (continued)

Member	Function
<b>ACBJBAIA</b>	Generate ISMF mountable tape volume list, save it in a table, and generate report from it
<b>ACBJBAIB</b>	Alter storage group volume status
<b>ACBJBAIC</b>	Test ACS routines
<b>ACBJBAID</b>	Generate mountable tape volume list and save it in a table
<b>ACBJBAIH</b>	Generate data set list using a previously saved query and save it in a table
<b>ACBJBAII</b>	Generate Aggregate Group list and report
<b>ACBJBAIJ</b>	Generate Aggregate Group list and save it
<b>ACBJBAIK</b>	Generate Aggregate Group report from a saved list
<b>ACBJBAIL</b>	Generate Data Class list and report
<b>ACBJBAIM</b>	Generate Data Class list and save it
<b>ACBJBAIN</b>	Generate Data Class report from a saved list
<b>ACBJBAIO</b>	Generate Management Class list and report
<b>ACBJBAIP</b>	Generate Management Class list and save it
<b>ACBJBAIQ</b>	Generate Management Class report from a saved list
<b>ACBJBAIR</b>	Generate mountable optical volume list and save it
<b>ACBJBAIS</b>	Generate mountable optical volume list and report
<b>ACBJBAIT</b>	Generate mountable optical volume report from a saved list
<b>ACBJBAIU</b>	Generate Storage Class list and report
<b>ACBJBAIV</b>	Generate Storage Class list and save it
<b>ACBJBAIW</b>	Generate Storage Class report from a saved list
<b>ACBJBAIX</b>	Generate Storage Group list and report
<b>ACBJBAIY</b>	Generate Storage Group list and save it
<b>ACBJBAIZ</b>	Generate Storage Group report from a saved list
<b>ACBJBAJ1</b>	Management class define/alter/display
<b>ACBJBAJ2</b>	Pool storage group define/alter
<b>ACBJBAJ3</b>	Tape storage group define/alter
<b>ACBJBAJ7</b>	Object backup storage group define/alter
<b>ACBJBAJ8</b>	VIO storage group define/alter
<b>ACBJBAJA</b>	Dummy storage group define/alter
<b>ACBJBAJB</b>	Object storage group define/alter
<b>ACBJBAJC</b>	Management Class Delete
<b>ACBJBAJD</b>	Data Class Delete
<b>ACBJBAJE</b>	Storage Class Delete
<b>ACBJBAJF</b>	Storage Group Delete

Table 30. SYS1.SACBCNTL Sample JCL Library Member List (continued)

Member	Function
<b>ACBJBAJG</b>	Aggregate Group Delete
<b>ACBJBAJH</b>	Copy pool backup storage group define/alter
<b>ACBJBAJI</b>	Generate a copy pool list and save it
<b>ACBJBAJJ</b>	Generate a copy pool list from a saved list
<b>ACBJBAJK</b>	Generate a copy pool list from a saved list, save it, and generate a report
<b>ACBJBAJL</b>	Copy pool delete
<b>ACBJBAJM</b>	Network Connection Delete
<b>ACBJBAK1</b>	CF Lock Set display / update
<b>ACBJBAK4</b>	Storage Class CF Lock display
<b>ACBJBAL1</b>	Tape Library define/alter/display
<b>ACBJBAL4</b>	Generate tape library list
<b>ACBJBAL5</b>	Generate tape library report from a previously saved list
<b>ACBJBAL6</b>	Generate tape library list, save it and generate report
<b>ACBJBAL7</b>	Tape Library delete
<b>ACBJBAM1</b>	Generate model command from saved ISMF table (data set list)
<b>ACBJBAM2</b>	Model commands from DCOLLECT data
<b>ACBJBAN1</b>	CF Cache Set display / update
<b>ACBJBAN4</b>	Storage Class CF Cache display
<b>ACBJBAOB</b>	Library allocation
<b>ACBJBAOD</b>	Generate data set report from a previously saved table (data set list)
<b>ACBJBAOF</b>	Generate volume report from a previously saved table (DASD volume list)
<b>ACBJBAOI</b>	ACS object information display
<b>ACBJBAOQ</b>	Translate ACS Routines
<b>ACBJBAOS</b>	Validate SCDS
<b>ACBJBAOT</b>	Generate tape report from a previously saved table (tape volume list)
<b>ACBJBAOU</b>	Generate data set list, save it in a table, and generate report from it
<b>ACBJBAOW</b>	Generate test cases from VMA extract file
<b>ACBJBAO7</b>	Generate DCOLLECT output
<b>ACBJBAP1</b>	Copy pool define/alter/display
<b>ACBJBAR2</b>	Generate SMS configuration report From DCOLLECT data
<b>ACBJBARD</b>	Generate data set report from DCOLLECT data
<b>ACBJBAS1</b>	Storage Class define/alter/display
<b>ACBJBAU2</b>	Update test cases test cases with expected results
<b>ACBJBAU4</b>	Job to customize the job card for all NaviQuest jobs

Table 30. SYS1.SACBCNTL Sample JCL Library Member List (continued)

Member	Function
ACBJBAXV	Generate DASD volume report from DCOLLECT data
ACBJBAX1	ACS cross reference Report

## JCL for ACBJBAOB

The following example shows the sample 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 DGTLLIB DATA SET; TABLE CAN BE LARGE - ALLOCATE A LARGE */
//* DATA SET. Be sure you change 'userid' to YOUR userid. */
//*-----*/
//*****
//*$MAC(ACBJBAOB) COMP(5695DF123): BATCH - CALLED PROC */
//* */
//* PROPRIETARY V3 STATEMENT */
//* LICENSED MATERIALS - PROPERTY OF IBM */
//*5694-A01 */
//* (C) COPYRIGHT 1998, 2008 IBM CORP. */
//* END PROPRIETARY V3 STATEMENT */
//* */
//* CHANGE ACTIVITY: */
//*$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 ISPLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST MESSAGES */
//*-----*/
//ISPLIB 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

```

## 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 31 on page 341](#) 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:

1. Do not modify these REXX EXECs and CLISTS.
2. 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 'xxxxxxx' command. Save the table, then use ISMF options to produce a flat file of the table for printing:
  - Option 11.7.1 for any table
  - Option 11.5.*n* for data set and volume tables.

Table 31. CLISTS and REXX EXECs for Storage Administration Tasks

Interactive ISMF Option	Storage Administration Task	CLIST or REXX EXEC
<b>1</b>	Generate data set list	ACBQBAI2
<b>2.1</b>	Generate DASD or optical device volume list	ACBQBAI4
<b>2.2</b>	Generate optical volume list	ACBQBAIE

Table 31. CLISTs and REXX EXECs for Storage Administration Tasks (continued)

<b>Interactive ISMF Option</b>	<b>Storage Administration Task</b>	<b>CLIST or REXX EXEC</b>
<b>2.3</b>	Generate tape volume list	ACBQBAI6
<b>3.1</b>	List management class	ACBQBAID
<b>3.2</b>	Display management class	ACBQBAJ1
<b>3.3</b>	Define management class	ACBQBAJ1
<b>3.4</b>	Alter management class	ACBQBAJ1
<b>4.1</b>	List data class	ACBQBAIC
<b>4.2</b>	Display data class	ACBQBAD1
<b>4.3</b>	Define data class	ACBQBAD1
<b>4.4</b>	Alter data class	ACBQBAD1
<b>5.1</b>	List storage class	ACBQBAIF
<b>5.2</b>	Display storage class	ACBQBAS1
<b>5.3</b>	Define storage class	ACBQBAS1
<b>5.4</b>	Alter storage class	ACBQBAS1
<b>5.5</b>	Cache display in storage class	ACBQBAS1
<b>5.6</b>	Lock display in storage class	ACBQBAS1
<b>6.1</b>	List storage group	ACBQBAIG



Table 31. CLISTs and REXX EXECs for Storage Administration Tasks (continued)

Interactive ISMF Option	Storage Administration Task	CLIST or REXX EXEC
<b>6.2</b>	Define dummy storage group	ACBQBAJA
	Define object backup storage group	ACBQBAJ7
	Define object storage group	ACBQBAJB
	Define pool storage group	ACBQBAJ2
	Define tape storage group	ACBQBAJ3
	Define VIO storage group	ACBQBAJ8
	Define copy pool backup storage group	ACBQBAJH
	Define Display Object type Storage Group	ACBSMDJ5
	Define Display Tape type Storage Group	ACBSMDJ5
<b>6.3</b>	Alter copy pool backup storage group	ACBQBAJH
	Alter dummy storage group	ACBQBAJA
	Alter object backup storage group	ACBQBAJ7
	Alter object storage group	ACBQBAJB
	Alter pool storage group	ACBQBAJ2
	Alter tape storage group	ACBQBAJ3
	Alter VIO storage group	ACBQBAJ8
	Alter Display Object type Storage Group	ACBSMDJ5
	Alter Display Tape type Storage Group	ACBSMDJ5
<b>6.4</b>	Storage group define/delete/alter volume	ACBQBAI9
<b>7.2</b>	ACS routine translate	ACBQBAO1
<b>7.3</b>	SCDS validation	ACBQBAO2
<b>7.4.3</b>	Test ACS Routines	ACBQBAIA
<b>7.5</b>	ACS object display	ACBQBAOI
<b>8.1</b>	Display base configuration	ACBQBAB1
<b>8.2</b>	Define base configuration	ACBQBAB1
<b>8.3</b>	Alter base configuration	ACBQBAB1
<b>8.6</b>	Cache display in base configuration	ACBQBAN1

Table 31. CLISTs and REXX EXECs for Storage Administration Tasks (continued)

<b>Interactive ISMF Option</b>	<b>Storage Administration Task</b>	<b>CLIST or REXX EXEC</b>
<b>8.7</b>	Cache update in base configuration	ACBQBAN1
<b>8.8</b>	Lock display in base configuration	ACBQBAK1
<b>8.9</b>	Lock update in base configuration	ACBQBAK1
<b>9.1</b>	List aggregate group	ACBQBAIB
<b>9.2</b>	Display aggregate group	ACBQBAA1
<b>9.3</b>	Define aggregate group	ACBQBAA1
<b>9.4</b>	Alter aggregate group	ACBQBAA1
<b>10.3.1</b>	List tape library	ACBQBAL4
<b>10.3.2</b>	Display tape library	ACBQBAL1
<b>10.3.3</b>	Define tape library	ACBQBAL1
<b>10.3.4</b>	Alter tape library	ACBQBAL1
<b>11.1.1</b>	Test cases from ISMF-saved list	ACBQBAG3
<b>11.1.2</b>	Test cases from DCOLLECT data	ACBQBAG1
<b>11.1.3</b>	Test cases from SMF data	ACSTST program
<b>11.1.4</b>	Test cases from VMA extract data	ACBQBAO3
<b>11.2</b>	ACS test listings comparison	ACBQBAC1
<b>11.3</b>	Enhanced ACS test listing	ACBQBAX1
<b>11.4</b>	Update test cases with expected test results	ACBQBAU1
<b>11.5.1</b>	Data set report from ISMF-saved list	ACBQBAR1
<b>11.5.2</b>	DASD volume report from ISMF-saved list	ACBQVAR1
<b>11.5.3</b>	Tape volume report from ISMF-saved list	ACBQBAR4

Table 31. CLISTs and REXX EXECs for Storage Administration Tasks (continued)

Interactive ISMF Option	Storage Administration Task	CLIST or REXX EXEC
<b>11.5.4</b>	Data set report from DCOLLECT data	ACBQBAR7
<b>11.5.5</b>	DASD volume report from DCOLLECT data	ACBQBAR6
<b>11.5.6</b>	SMS configuration report from DCOLLECT data	ACBQBAR8
<b>11.6.1</b>	Model command from ISMF-saved list	ACBQBAM1
<b>11.6.2</b>	Model command from DCOLLECT data	ACBQBAM2
<b>C.1</b>	List copy pool	ACBQBAIJ
<b>C.2</b>	Display copy pool	ACBQBAP1
<b>C.3</b>	Define copy pool	ACBQBAP1
<b>C.4</b>	Alter copy pool	ACBQBAP1
<b>S.1</b>	List network connection	ACBQBAIK
<b>S.2</b>	Display network connection	ACBQBAC3
<b>S.3</b>	Define network connection	ACBQBAC3
<b>S.4</b>	Alter network connection	ACBQBAC3

## 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 “Sample JCL (ACBJBAI1) for generating a data set list ” on page 348 for the sample JCL and parameters.

Use the parameters in Table 32 on page 345 with ACBQBAI2. At least one OP and one value should be included when specifying a parameter.

Table 32. Using ACBQBAI2, ACBJBAI7, and ACBJBAOU Parameters

Parameters	Description
<b>ALLOCSP(OP1 nnn1 BOOL OP2 nnn2)</b>	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

Table 32. Using ACBQBAl2, ACBJBAI7, and ACBJBAOU Parameters (continued)

Parameters	Description
<b>ALLOCUT(OP dd1 dd2 dd3 dd4)</b>	Where OP is EQ or NE; allocation unit, BLK/TRK/ABS/CYL/MB/KB/BYT
<b>BLKSIZE(OP1 nnn1 BOOL OP2 nnn2)</b>	Where 'OP1' and 'OP2' are EQ, GT, LE, or NE; 'nnn1' and 'nnn2' are the block size values; BOOL is AND or OR
<b>BLKUNUSED(OP1 nnn1 BOOL OP2 nnn2)</b>	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
<b>CATNAME(catalog name)</b>	Where 'catalog name' is the name of the catalog to be searched for the dsns
<b>CATVOL(DDDDDD)</b>	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
<b>CCSIDSC(OP dd1 dd2 dd3 dd4)</b>	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
<b>CFCSTNM(OP dd1 dd2 dd3 dd4)</b>	Where OP is EQ or NE; CF cache structure name
<b>CFCTNM(OP dd1 dd2 dd3 dd4)</b>	CF cache set name
<b>CHGIND (OP DD1 DD2 DD3 DD4)</b>	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO for change indicator bit setting for the DSN
<b>CFMOST(OP dd1 dd2 dd3 dd4)</b>	Where 'OP' is EQ or NE; CF monitor status ON/OFF
<b>CFSTIND(OP dd1 dd2 dd3 dd4)</b>	Where 'OP' is EQ or NE; CF status indicator CE/CQ/VRRQ/VRLS/VQ
<b>COMPFMT (OP DD1 DD2 DD3 DD4)</b>	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO for compressed format for the data set
<b>CREATEDT (OP1 dat1 BOOL OP2 dat2)</b>	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
<b>DATACLS(OP DDD1 DDD2 DDD3 DDD4)</b>	Where 'OP' is EQ or NE, and 'DDD1' through DDD4' are the data classes
<b>DDMATTR(OP dd1 dd2 dd3 dd4)</b>	Where 'OP' is EQ or NE; DDM attributes YES/NO
<b>DEVTYPE(OP DDD1 DDD2 ... DDD8)</b>	Where 'OP' is EQ, NE, GT, LE, and so on, and 'DDD1' through DDD8 are the device types
<b>DSENV(OP dd1 dd2 dd3 dd4)</b>	Where 'OP' is EQ or NE; dataset environment MANAGED/UNMANAGED
<b>DSN(dsname)</b>	Where dsname is the name or generic name of each data set to be listed. ISMF uses dsname to generate lists of data set(s) and volume(s). The default of dsname is '*'
<b>DSNTYP(OP dd1 dd2 dd3 dd4)</b>	Where 'OP' is EQ or NE; DS name type EXTENDED/HFS/LIBRARY/OTHERS
<b>DSORG(OP DD1 DD2 ... DD8)</b>	Where 'OP' is EQ or NE, and 'DD1' through 'DD8' are the data set organizations
<b>ENTRYTYP(OP DDD1 DDD2 ... DDD12)</b>	Where 'OP' is EQ or NE, and 'DDD1' through 'DDD12' are the entry types (for example, DEFERRED, AIX®, CLUSTER, and GDG)
<b>EXPIREDT (OP1 dat1 BOOL OP2 dat2)</b>	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
<b>HSMDATA(Y N)</b>	This specifies whether or not the user wants to use the ACQUIRE DATA FROM DFHSM option. The default is 'N'.
<b>LASTBKUP (OP1 dat1 BOOL O2 dat2)</b>	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
<b>LASTREF (OP1 dat1 BOOL OP2 dat2)</b>	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

Table 32. Using ACBQBAI2, ACBJBAI7, and ACBJBAOU Parameters (continued)

Parameters	Description
<b>LRECL(OP1 nnn1 BOOL OP2 nnn2)</b>	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
<b>LISTTYP(I E)</b>	Inclusive/Exclusive or I/E Inclusive: Display list by Inclusive criteria Exclusive: Display list by Enclusive criteria
<b>MGMTCLS(OP nnnnnn1 ... Nnnnnn4)</b>	Where 'OP' is EQ or NE, and 'nnnnnn1' through 'nnnnnn4' are the management classes
<b>MULTVOL (OP DD1 DD2 DD3 DD4)</b>	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are either YES or NO if data set is multivolume
<b>NOTUSED%(OP1 nn1 BOOL OP2 nnn2)</b>	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
<b>NUMEXT(OP1 nnn1 BOOL OP2 nnn2)</b>	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
<b>NUMSTRIPE (OP1 nnn1 BOOL OP2 nnn2)</b>	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
<b>OPTIMAL (OP1 nnn1 BOOL OP2 nnn2)</b>	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.
<b>OWNER (OP DD1 DD2 DD3 DD4)</b>	Where 'OP' is EQ or NE, and 'DD1' thru 'DD4' are owners of the the datasets.
<b>QSAVE(nnnnnnnn)</b>	Where 'nnnnnn' is the query name to be created with all saved variables
<b>QUERY(nnnnnnnn)</b>	Where 'nnnnnn' is the query name to be used for all the variables
<b>REBLOCK(OP DDD1 DDD2 ... DDD3)</b>	Where 'OP' is EQ and and 'DDD1' through 'DDD3' are either YES or NO.
<b>RECFMT(OP DDD1 DDD2 .. DDD8)</b>	Where 'OP' is EQ or NE, and 'DDD1' through 'DDD8' are the record formats. List built from the specified criteria.
<b>SECALLOC (OP1 nnn1 BOOL OP2 nnn2)</b>	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.
<b>SOURCEGL(1 2)</b>	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.
<b>SOURCENL(1 2)</b>	Specifies whether the generated list comes from 1 - VTOC , or 2 - Catalog. The default is '2'.
<b>STORCLS(OP nnnnnn1 .. Nnnnnn4)</b>	Where 'OP' is EQ or NE, and 'nnnnnn1' through 'nnnnnn4' are the storage classes
<b>STORGRP(NNNNNN)</b>	Where NNNNNN is a fully specified storage group.
<b>USEDSPC(OP1 nnn1 BOOL OP2 nnn2)</b>	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
<b>USERDATAREDUCT% (OP1 nnn1 BOOL OP2 nnn2)</b>	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

Table 32. Using ACBQBAI2, ACBJBAI7, and ACBJBAOU Parameters (continued)

Parameters	Description
<b>VTOCDATA(Y N)</b>	Specifies whether the user desires the ACQUIRE DATA FROM VOLUME OPTION. The default is 'N'.
<b>VTOCVSR(VVVVVV)</b>	Specifies the volsers whose VTOCs are to be searched. The user may specify from 1–6 alphanumeric characters and an asterisk for filtering.

### 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.

### Sample JCL (ACBJBAI1) for generating a data set list

The following example shows the sample JCL for ACBJBAI2.

```

//*****
//*****
//*
//* 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/3490/3590-1/3592 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 Exclusive criteria      *
/*      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      *
/*      STORGRP - STORAGE GROUP 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 ACBQBAOB,
//      PLIB1=SYS1.DGTPLIB,
//      TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAI2 SAVE DSNLIST +
VTOCDATA(Y) HSMDATA(Y) +
ALLOCS(PT 500) DSORG(NE PS) +
DSN('IBMUSER.**')) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****

```

## 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 [“Sample JCL \(ACBJBAI4\) for generating a volume list ” on page 352.](#)

For ACBQBAI4, use the parameters listed in [Table 33 on page 349.](#) At least one OP and one value should be included when specifying a parameter.

Table 33. ACBQBAI4 Parameters

Parameter	Description
<b>ALLOCS(PT1 nnn1 BOOL OP2 nnn2)</b>	Specifies the amount of allocated space. Accepted values for nnn1 are from 0 to 99999999 (in KB or MB); '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.

Table 33. ACBQBAI4 Parameters (continued)

Parameter	Description
<b>ALSPTRK(OP1 nnn1 BOOL OP2 nnn2)</b>	Specifies the amount of track-managed allocated space. Accepted values for nnn1 and nnn2 are from 0 to 99999999 (in KB or MB); 'OP1' and 'OP2' can be EQ, NE, GT, GE, LT or LE; 'BOOL' is AND or OR.
<b>CDSNAME(cds.name)</b>	The SCDS that extracts the volume information from for the volume list
<b>CFVOLST(OP nn1 nn2 nn3 nn4)</b>	'OP' is EQ or NE; CF volume status ENABLED/QUIESCING/QUIESCED
<b>CFWSTAT(OP nnn1 nnn2 nnn3 nnn4)</b>	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.
<b>DEVTYPE(xxxxxxxx)</b>	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 xx is the generic device type.
<b>DFWSTAT(OP1 nnn1 nnn2 nnn3 nnn4)</b>	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
<b>DUPLXSTAT(OP1 nnn1 nnn2 nnn3 nnn4)</b>	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
<b>FRAG(OP1 nnn1 BOOL OP2 nnn2)</b>	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
<b>FREEDSCB(OP1 nnn1 BOOL OP2 nnn2)</b>	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.
<b>FREESPC(OP1 nnn1 BOOL OP2 nnn2)</b>	Specifies the amount of free space. Accepted values for 'nnn1' are from 0 to 99999999 (in KB or MB); '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.
<b>FREESPC%(OP1 nnn1 BOOL OP2 nnn2)</b>	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.
<b>FREEVIR(OP1 nnn1 BOOL OP2 nnn2)</b>	Specifies the amount of free VIRs. Accepted values for 'nnn1' are from 0 to 999999; '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.
<b>FREEXT(OP1 nnn1 BOOL OP2 nnn2)</b>	Specifies the number of free extents. Accepted values for 'nnn1' are from 0 to 99999999; '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.
<b>FRSPTRK (OP1 nnn1 BOOL OP2 nnn2)</b>	Specifies the amount of track-managed free space. Accepted values for nnn1 and nnn2 are from 0 to 99999999 (in KB or MB); 'OP1' and 'OP2' can be EQ, NE, GT, GE, LT or LE; 'BOOL' is AND or OR.
<b>FROMDEV(nnn)</b>	First device number in the range to be listed in the volume list. Used with the LASTDEV parameter.
<b>INIASRES (OP1 nnn1 nnn2 nnn3 nnn4)</b>	Limits the volumes included in the list to those initialized as reserved. Accepted values for nnn# are YES or NO; 'OP' is EQ or NE.
<b>INDEX(OP nnn1 nnn2 nnn3 nnn4)</b>	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.



Table 33. ACBQBAI4 Parameters (continued)

Parameter	Description
<b>INDEX%USED (OP1 nnn1 BOOL OP2 nnn2)</b>	Limits the volumes included in the list to those with specified proportions of Index % used. Accepted values for 'nnn1' are from 0 to 11, 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.
<b>LASTDEV(nnn)</b>	Last device number in the range to be listed in the volume list. Used with the FROMDEV parameter.
<b>LGEXTRK(OP1 nnn1 BOOL OP2 nnn2)</b>	Limits the volumes in the list to those with largest track-managed extents. Accepted values for 'nnn1' and 'nnn2' are from 0 to 99999999 (in KB or MB); 'OP1' and 'OP2' can be EQ, NE, GT, LE, LT and GE; 'BOOL' is AND or OR.
<b>LISTTYP(I E)</b>	Inclusive/Exclusive or I/E Inclusive: Display list by Inclusive criteria Exclusive: Display list by Exclusive criteria
<b>LRGEXT(OP1 nnn1 BOOL OP2 nnn2)</b>	Limits the volumes in the list to those with largest extents of a specified size. Accepted values for 'nnn1' are from 0 to 99999999 (in KB or MB); '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.
<b>OTHERDEV(OP1 nnn1 BOOL OP2 nnn2)</b>	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.
<b>OWNERID(OP nnn1 nnn2)</b>	Limits the volumes included in the list to those with the specific ID. Accepted values for nnn# are any 1 to 14 character string.
<b>PHYDATA(Y N)</b>	Specifies whether or not the user desires the ACQUIRE PHYSICAL DATA OPTION. The default is 'N'.
<b>PHYSTAT(OP nnn1 nnn2 nnn3 nnn4)</b>	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.
<b>QSAVE (nnnnnnnn)</b>	Where 'nnnnnnnn' is the query name to be created with all saved variables.
<b>QUERY (nnnnnnnn)</b>	Where 'nnnnnnnn' is the query name to be used for all variables.
<b>RDCACHE(OP1 nnn1 nnn2 nnn3 Nnn4)</b>	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.
<b>SHRDASD(OP nnn1 nnn2 nnn3 Nnn4)</b>	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.
<b>SOURCEGL(1 2)</b>	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'.
<b>SOURCENL(1 2)</b>	Specifies whether the generated list comes from PHYSICAL(1) or SMS(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.
<b>SPCDATA(Y N)</b>	Specifies whether or not the user desires the ACQUIRE SPACE DATA OPTION. The default is 'N'.
<b>STORGRP(XXXXXXXX)</b>	This is the STORGRP to extract the volume information from.
<b>SUBSYSID(OP1 nnn1 BOOL OP2 nnn2)</b>	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.
<b>USEATTR(OP nnn1 nnn2 nnn3 nnn4)</b>	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.

Table 33. ACBQBAI4 Parameters (continued)

Parameter	Description
<b>VOL (list of VOLS)</b>	Where "list of VOLS" is the format used by ISMF for generating lists of data sets and volumes. The default is '*1'.
<b>VOLSTYLE(1 2 3)</b>	Specifies whether the generated list will come from ONLINE(1), NOT ONLINE(2), or EITHER(3). The default is '1'.
<b>VTOC%USED(OP1 nnn1 BOOL OP2 nnn2)</b>	Limits the volumes included in the list to those with specified proportions of VTOC % used. Accepted values for 'nnn1' are from 0 to 11, 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.

### Sample JCL (ACBJBAI4) 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: *
//* *
//* PARAMETER FOLLOWING SAVE - NAME OF SAVED LIST (OUTPUT) *
//* ALLOCSP - 0 to 99999999 followed by K (kilobytes) or *
//* M (megabytes). If K or M aren't specified *
//* the value will be defaulted to K. *
//* ALSPTRK - 0 to 99999999 followed by K (kilobytes) or *
//* M (megabytes). If K or M aren't specified *
//* the value will be defaulted to K. *
//* 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 *
//* FRAG - 1 to 999 *
//* FREEDSCB - 0 to 99999 *
//* FREESPC - 0 to 99999999 followed by K (kilobytes) or *
//* M (megabytes). If K or M aren't specified *
//* the value will be defaulted to K. *
//* FRSPTRK - 0 to 99999999 followed by K (kilobytes) or *
//* M (megabytes). If K or M aren't specified *
//* the value will be defaulted to K. *
//* FREESPC% - 0 to 100 (with no % sign) *
//* FREEVIR - 0 to 99999 *
//* FREEEXT - 0 to 9999999 *
//* FROMDEV - 1 to 4 HexaDecimal digits *
//* INDEX - DISABLED/ENABLED/NONE *
//* INIASRES - Initialized as Reserved (YES / NO) *
//* INDEX%USED - 0 to 100 (with no % sign) *
//* LASTDEV - 1 to 4 HexaDecimal digits *
//* LISTTYP - Inclusive/Exclusive or I/E *
//* Inclusive: Display list by Inclusive criteria *
//* Exclusive: Display list by Enclusive criteria *
//* LRGEXT - 1 to 99999999 followed by K (kilobytes) or *
//* M (megabytes). If K or M aren't specified *
//* the value will be defaulted to K. *
//* LGEXTRK - 1 to 99999999 followed by K (kilobytes) or *
//* M (megabytes). If K or M aren't specified *
//* the value will be defaulted to K. *
//* OTHERDEV - 1 to 4 Hex Decimal digits *
//* OWNERID - Owner ID, 1 to 14 characters *
//* 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     - Storage 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) *
//*  VTOC%USED   - 0 to 100 (with no % sign)              *
//*                                                     *
//*****
//DASDLST EXEC ACBJBAOB,
//          PLIB1=SYS1.DGTPLIB,
//          TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER) MSGID
ISPSTART CMD(ACQBBAI4 +
SAVE CHKFRAG QSAVE(CHKFRAG) +
SPCDATA(Y) PHYDATA(Y) +
VOL(TSC*) FRAG(GT 450)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//
*****

```

## Generate a tape volume list: ACQBBAI6

ACQBBAI6 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 [“Sample JCL \(ACBJBAIA\) for generating a tape list” on page 353](#).

For ACQBBAID, use the parameters listed in [Table 34 on page 353](#).

Table 34. ACQBBAID Parameters

Parameter	Description
<b>LIBNAME (nnn)</b>	Specifies the library name to query against. The default is '*'.
<b>SOURCEGL(1 2)</b>	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'.
<b>STORGRP(XXXXXXXX)</b>	Specifies the storage group to extract the volume information from. The default is '*'.
<b>VOL(list of VOLS)</b>	Where 'list of VOLS' is in the format for ISMF. The default is '*'.

## Sample JCL (ACBJBAIA) 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 ACBJBA0B,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 ISPFIL, WHERE THE GENERATED REPORT IS SAVED *
//NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY *
//      DATA SET. *
//          *
//*****
//ALCISPFL EXEC PGM=IEFBR14
//ISPFIL DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),
//      BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA

//*****
//          *
//STEP TO GENERATE THE TAPE VOLUME REPORT FROM THE SAVED LIST *
//          *
//PARAMETER FOR ACBQBAR4 - SAVED LIST NAME (FROM GENLIST STEP) *
//ISPFIL - TAPE VOLUME REPORT (OUTPUT, FROM ALCISPFL STEP) *
//SYSDA - KEY WORDS SPECIFYING COLUMNS TO BE INCLUDED IN THE *
//      GENERATED REPORT *
//          *
//*****
//GENREP EXEC ACBJBA0B,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR4 TAPELIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//ISPFIL DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=OLD
//SYSDA 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=*
//SYSDA DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****

```

## Define/Alter/Display management class: ACBQBAJ1

SYS1.SACBCNTL member ACBJBAJ1 calls ACBQBAJ1 to define, alter, or display data set management class in batch.

See the following example 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/alterd, not exceeding 120 chars. *  
//*  
//* EXPNOUSE : The datasets will expire if they are not used for *  
//* the number of days specified here. *  
//* @D1C *  
//* Possible values 1 - 93000, 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. @D1C *  
//* Possible values 0 - 93000, YYYY/MM/DD or NOLIMIT. *  
//*  
//* RETNLIM : Possible values 0 - 93000, NOLIMIT. @D1C *  
//* 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 - 93000 -> Use this value only if the RETPD or *  
//* EXPDT is more than this limit. @D1C *  
//* NOLIMIT -> Do not set a limit to RETPD or EXPDT. *  
//*  
//* 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 DFSMSshm
/**                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 field 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      ->
/**
/**
/** L2DNOUSE : Use this field to specify whether data sets 13@G3A
/**                will migrate to cloud storage. It is to be managed
/**                by DFSMSshm. Possible values are,
/**
/**      0      -> Data sets will migrate directly to
/**                to cloud storage.
/**
/**      NOLIMIT    -> Data that migrates to ML2 will not
/**                further migrate to cloud storage.
/**
/**      BLANK      -> Default (NOLIMIT) is assumed.
/**
/**
/** 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.
/**
/**
/**
/**
/** SIZELTE : Use this field to specify the Low data set 9@G3A
/**                size threshold when data sets will be processed
/**                according to the actions, defined in field ACTLTE.
/**                Possible values are,
/**
/**      1 to 4294967295 -> Data set size in tracks.
/**
/**      BLANK      ->
/**
/**
/** ACTLTE : Use this field to specify which action to 13@G3A
/**                perform if the data set size is less than or equal
/**                to the given size, defined in field SIZELTE.
/**                Possible values are,
/**
/**      NONE      -> No action is taken.
/**
/**      ML1      -> Target migration level is ML1.
/**
/**      ML2      -> Target migration level is ML2.
/**
/**      MIG      -> Target migration level is ML1 or ML2.
/**
/**      TRANS     -> Data Set transition.
/**
/**      CLOUD     -> Target migration level is CLOUD.
/**
/**      BLANK     ->
/**
/**
/**
/** SIZEGT : Use this field to specify the High data set 9@G3A
/**                size threshold when data sets will be processed
/**                according to the actions, defined in Action (GT).
/**                Possible values are,
/**
/**      1 to 4294967295 -> Data set size in tracks.
/**
/**      BLANK      ->
/**
/**

```

```

/** ACTGT : Use this field to specify which action to 13@G3A*
/** perform if the data set size is greater than *
/** the given size, defined in field SIZEGT. *
/** Possible values are, *
/** *
/** NONE -> No action is taken. *
/** ML1 -> Target migration level is ML1. *
/** ML2 -> Target migration level is ML2. *
/** MIG -> Target migration level is ML1 or ML2. *
/** TRANS -> Data Set transition. *
/** CLOUD -> Target migration level is CLOUD. *
/** BLANK -> *
/** *
/** 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 - 999 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 -> *
/** *
/** 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 -> Without the concurrent copy technique. *
/** VR -> Virtual concurrent copy technique 9@AOA *
/** must be used. *
/** VP -> Virtual concurrent copy technique *
/** should be used. *
/** CR -> Cache-based concurrent copy technique *
/** must be used. *
/** CP -> Cache-based concurrent copy technique *
/** should be used. *
/** *
/** 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 *
/** Default: FIRST is assumed if PQUAIMTH is 2@F1A *
/** specified. *
/** *
/** PQUAIMTH : Month of each quarter the CT occurs. *
/** Possible values, 1 - 3, or BLANK *
/** *
/** PYRLDAY : The day of each year the CT occurs. *
/** Possible values, 1 - 366, FIRST, LAST or BLANK *
/** Default: FIRST is assumed if PYRLIMTH is 2@F1A *
/** specified. *
/** *
/** PYRLIMTH : The month of each year the CT occurs. *
/** Possible values, 1 - 12, or BLANK *
/** *
/** VERSIONS : Specify how many versions of an aggregate group *
/** associated with the management class are to be *
/** maintained. Possible values are 1 - 9999, NOLIMIT 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 -> Without the concurrent copy technique. *
/** VR -> Virtual concurrent copy technique 9@AOA *
/** must be used. *
/** VP -> Virtual concurrent copy technique *
/** should be used. *
/** CR -> Cache-based concurrent copy technique *
/** must be used. *
/** CP -> Cache-based concurrent copy technique *
/** should be used. *
/** *
/** TRCPYTECH : Specifies which copy technique should be used 9@E1A*
/** for the class transition of data associated with *
/** this management class. Possible values are *
/** FRP -> FR PREFERRED. *
/** FRR -> FR REQUIRED. *
/** STD -> STANDARD. *
/** PMP -> FC PRESMIRREF. *
/** PMR -> FC PRESMIRREQ. *
/** FCX -> FC XRCPRIMARY - can be specified only 6@G4A*
/** if RPFC for XRC support is enabled *
/** (FLASHCOPYTOXRC=YES parameter is set in *
/** the DEVSUPxx PARMLIB member and activated). *
/** For more information, see "z/OS DFSMS *
/** Advanced Copy Services". *
/** *
/** SERIALERREX: Specifies the database/exit to invoke when 5@E1A*
/** there is a serialization error. Possible values are *
/** DB2, CICS, ZFS, EXIT -> invokes an user exit, or *
/** NONE. *
/** *
/** CLOUDNAME: Specifies the name of a previously defined 13@G3A*
/** or network connection construct for the data set 6@H1C*
/** CLNCNAME migration to the cloud storage during automatic *
/** migration (Primary Space Management, Interval *
/** and On-demand migration). CLOUDNAME and CLNCNAME *
/** keywords have the same meaning and can be used *
/** interchangeably. *
/** *
/** 1 to 30 alphanumeric or national characters starting*
/** with alphabetic or national character *
/** (@, #, $). *
/** *
/** BLANK -> *
/** *
/** RETMETHOD: Specifies how new tape volumes are to be 61@G1A*
/** managed by DFSMSrmm. The retention method *
/** for a volume set is chosen when its first *
/** file is written. Possible values are, *
/** EXPDT -> Retention and expiration is managed *
/** based on expiration date, date of *
/** last reference, or catalog status. *
/** VRSEL -> Policy management is to be managed *
/** by DFSMSrmm based on VRSEs using *
/** existing, supported combinations *
/** of data set name, jobname and the *
/** MC name. *
/** blank -> Value is not specified. *
/** VOLSETMGL: Specifies how DFSMSrmm is to retain volumes *
/** or multivolume sets that are managed by the *
/** EXPDT retention method. Possible values are, *
/** VOLUME -> The expiration date is determined *
/** separately for each volume in the *
/** set. Unless defined differently, *
/** the expiration date is the maximum *
/** of all data set expiration dates on *
/** the volume. Each file on a volume *
/** can increment the volume expiration *

```

```

/**          date. All files of a multivolume data      *
/**          set have the same expiration date.         *
/**          FIRSTFILE -> the expiration attributes of the first *
/**          file are used for the whole volume         *
/**          set.                                         *
/**          SET    -> the highest data set expiration date *
/**          set for all files on a tape volume set     *
/**          determines the expiration date for all    *
/**          volumes in the set.                         *
/**          blank -> Value is not specified.           *
/**          *                                           *
/** EXCLVRSEL: Specifies that any tape data sets created on a *
/**          tape volume set which is managed by the VRSEL *
/**          retention method can be excluded from VRSEL *
/**          inventory management. The data set VRSELEXCLUDE *
/**          attribute is set for all data sets on volumes *
/**          managed by the EXPDT retention method and is not *
/**          affected by this support. Possible values are, *
/**          Y -> Yes, if the data set is on a volume     4@G2C*
/**          managed by the VRSEL retention method, the *
/**          data set is excluded from VRSEL inventory *
/**          management processing.                      *
/**          N -> No, no exclusion, used only for data    3@G2C*
/**          sets on a volume managed by the VRSEL      *
/**          retention method.                           *
/**          blank -> Value is not specified.           *
/**          *                                           *
/** WHILECTLG: Specifies the default WHILECATALOG value for data *
/**          sets on volumes managed by the EXPDT retention *
/**          method. Possible values are,                *
/**          OFF -> The catalog status of the data set has no *
/**          effect on when it expires.                  *
/**          ON  -> The data set will be retained past its *
/**          expiration date if it is cataloged.          *
/**          UNTILEXPIRED -> The data set will expire after *
/**          its status changes to uncataloged            *
/**          or if the expiration date is reached.        *
/**          blank -> Value is not specified.           *
/**          *                                           *
/**          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=(MOD,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() +                                  *
/**          EXPDTE() +                                    *
/**          RETNLIM() +                                   *
/**          PARTREL() +                                   *
/**          PRINOUSE() +                                  *
/**          LVINOUSE() +                                  *
/**          L2DNOUSE() +                                  *
/**          CMDORAUT() +                                  *
/**          SIZELTE() +                                   *
/**          ACTLTE() +                                    *
/**          SIZEGT() +                                    *
/**          ACTGT() +                                     *
/**          PRIGDDEL() +                                  *
/**          GDGROLL() +                                   *
/**          BACKUPFR() +                                  *
/**          NUMBKDSE() +                                  *
/**          NUMBKDSD() +                                  *
/**          RETDYDSD() +                                  *
/**          RETDYEXT() +

```

```

CMDBKUP() +
AUTOBKUP() +
BKUPTTECH() +
TMSCYRS() +
TMSCMTH() +
TMSCDYS() +
TMSLUYRS() +
TMSLUMTH() +
TMSLUDYS() +
PMTHODAY() +
PQUAODAY() +
PQUAIMTH() +
PYRLDAY() +
PYRLIMTH() +
VERSIONS() +
RTNOVERS() +
RTOVUNIT() +
RTNEVERS() +
RTEVUNIT() +
CPYSERLN() +
ACPYTECH() +
TRCPYTECH() +
SERIALERREX() +
CLNCNAME() +
RETMETHOD() +
VOLSETMGL() +
EXCLVRSEL() +
WHILECTLG() +
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() +
)
/*

```

## 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 [“Sample JCL \(ACBJBAIB\) for storage group volume add/Delete” on page 362](#) 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 (ACBJBAIB) for storage group volume add/Delete

To delete a volume, specify the following:

- SCDSNAME
- SG
- VOL
- VOLDEL for the DDNAME

Following is the sample JCL for ACBJBAIB.

```
//*****
//*
//* SAMPLE JCL TO ADD NEW VOLUMES AND THEIR STATUS
//*
//* INSTRUCTIONS BEFORE SUBMITTING:
//*
//*   CHANGE JOBCARD
//*   CHANGE PREFIX
//*   CHANGE PARAMETERS
//*
//*   TEMPFIL - 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*
//*                                     13A@WA53756*
//*   STATUSALL - STATUSALL (ENABLE/NOTCON/DISALL/DISNEW/QUIALL/
//*                 QUINEW)
//*           If Volume status in all the Systems needs to be
//*           set to a single value (for example ENABLE),
//*           STATUSALL is an easier option compared to the
//*           parameter STATUS.
//*
//*   Note: STATUSALL and STATUS are mutually exclusive.
//*         And so, while specifying value for one of these
//*         parameters, either the other parameter should
//*         not be specified or if specified, it should not
//*         have any value specified.
//*
//*   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*
//*
//* 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(ACQBQBAI9) +
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,QUINNEW,ENABLE,NOTCON,QUIALL,DISNEW,DISALL,QUIALL,+
DISNEW,ENABLE,ENABLE,DISNEW,QUIALL,DISALL,DISALL,NOTCON,+
NOTCON,QUINNEW,ENABLE,NOTCON,QUIALL,DISNEW,ENABLE,QUIALL)
SCDS(MYCDS) VOL(SMS) FROM(14) TO(20) SUF(1) TYPE() SG(PRIMARY) STATUS(ENABLE)
SCDS(MYCDS) VOL(SMSA) FROM(A) TO(X) SUF(Y) TYPE(A) SG(PRIMARY) STATUS(ENABLE)
SCDS(MYCDS) VOL(SMSX) FROM(1) TO(C) SUF(G) TYPE(X) SG(PRIMARY) STATUS(ENABLE)
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) STATUSALL(ENABLE)
SCDSNAME(MYCDS) VOL(SMS912) SG(PRIMARY) STATUS(ENABLE) STATUSALL()
SCDSNAME(MYCDS) VOL(SMS913) SG(PRIMARY) STATUS() STATUSALL(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)
/******

```

## ACS routine translate: ACQBQBAO1

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. ACQBQBAO1 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 [“Sample JCL \(ACBJBAC2\) for ACS routine translate”](#) on page 363 for the sample JCL and parameters.

## Sample JCL (ACBJBAC2) for ACS routine translate

Following is the sample JCL for ACS routine translate.

```

/******
/*
/* SAMPLE JCL TO DO FOUR OPERATIONS:
/*
/*

```

```

//*      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 ACBQBAOB,
//              PLIB1='SYS1.DGTPLIB',
//              TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
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 *
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=*
//*****
//* 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(%ACQBAC1 +
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=*
//*

```

## 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 62 on page 367](#) 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)
//*     UPDHLVLS - 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 UPDHLVLS 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(ACBQBA02 +
SCDSNAME(DSNREPN) TYPE(*) +
LISTNAME(VALIDAT.LISTING) +
UPDHLVLS( )) +
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 62. 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 63 on page 368](#) 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 63. 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 64 on page 369](#) for the sample JCL and parameters.

### Recommendations and Restrictions:

1. 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.
2. 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.
3. 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 ACBJBA0B,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACQBAG3 TABLENM(DSNLIST) +  
INTEST(NO) PRFX(TEST) +  
TESTPDS('IBMUSER.TESTCASE.LIBRARY') REPLACE(Y)) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//*****
```

Figure 64. 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 65 on page 369](#) 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 ACBJBA0B,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACQBAG1 'IBMUSER.DCOLLECT.DATA' 10 TEST +  
'IBMUSER.TESTCASE.LIBRARY' Y) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//*****
```

Figure 65. 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 66 on page 370](#) 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 66. 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 67 on page 370](#) 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 ACBJBAO3,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) BDISP(99999999)  
/*
```

Figure 67. 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 68 on page 371](#) 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 ACBJBA0B,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
//  PROFILE PREFIX(IBMUSER)
//  DEL COMPARE.LISTING
//  DEL TESTCASE.EXCP

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 68. Sample JCL for ACBJBAC1

## 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 [“Sample JCL \(ACBJBAX1\) for generating an enhanced ACS test listing”](#) on page 372 for the sample JCL and parameters.

Use the following parameters with the ACBQBAX1:

Parameters	Description
<b>DSNINFO(YIN)</b>	Includes the data set name in the cross-reference
<b>UNITINFO(YIN)</b>	Includes the unit name in the cross-reference
<b>SIZEINFO(YIN)</b>	Includes size information (in K) in the cross-reference
<b>EXPTINFO(YIN)</b>	Includes expiration date information in the cross-reference
<b>JOBINFO(YIN)</b>	Includes jobname information in the cross-reference
<b>PGMINFO(YIN)</b>	Includes program name information in the cross-reference

### Sample JCL (ACBJBAX1) 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
//**
//*****
//** 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=*
//*

```

## 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 69 on page 373](#) 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 69. 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 [“Sample JCL \(ACBJBAOD\) for generating a report from ISMF-saved data set list” on page 375](#) 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
<b>%NOTUSED</b>	Prints the amount of space <i>not</i> used for the data set.
<b>ALLOCSP ALLOCSPC</b>	Prints the data set's allocated space.
<b>ALLOCUSED</b>	Prints the amount of used space for the data set.
<b>BACKUP LASTBKUP</b>	Prints the date of the last backup for the data set.
<b>BLKSIZE CISIZE</b>	Prints the BLKSIZE or CISIZE of the data set.
<b>BLKUNUSED</b>	Prints the number of unused blocks for the data set.
<b>CCSIDDES</b>	CCSID description
<b>CFMONST</b>	CF monitor status
<b>CHANGE CHGIND</b>	Is the change indicator set for the data set?
<b>CFSTATUS</b>	CF status indicator
<b>COMPRESS</b>	Compressed format
<b>CREATE CREATEDT</b>	Prints the data set's create date.
<b>CSETNAME</b>	CF cache set name
<b>CSTRNAME</b>	CF cache structure name
<b>DATACLAS DC</b>	Prints the data class for the data set, if there is one.
<b>DDMATTR</b>	DDM attribute
<b>DEVTYPE DEVICE</b>	Prints the device type that the data set resides on.
<b>DSNAMETY</b>	Dataset name type
<b>DSNAME DSN</b>	Prints the data set name.
<b>DSNLENGTH=nn</b>	Limits print of data set name to nn characters. Defaults to 44 characters.
<b>DSORG</b>	Prints the data set's organization.
<b>ENTRYTYPE ENTRY</b>	Prints the entry type of data set.
<b>ENVIRONMENT ENVIR</b>	Prints the data set's environment, managed or unmanaged.
<b>EXPIRE EXPIREDT</b>	Prints the data set's expiration date.
<b>EXTNUM</b>	Prints the data set's extension number.
<b>LASTREF LASTREFDT</b>	Prints the last reference date of the data set.
<b>LRECL</b>	Prints the LRECL of the data set.
<b>MGMTCLAS MC</b>	Prints the management class for the data set, if there is one.
<b>MULTVOL</b>	Is the data set a part of a multivolume data set?
<b>NUMEXT EXTNUM</b>	Prints the number of extents for the data Set.
<b>NUMSTR</b>	Number of stripes
<b>OPTIMAL</b>	Prints the optimal block size of the data Set.
<b>OWNER</b>	Prints the owner of the data set, if there is one.
<b>PAGELength=nn</b>	Sets page length for the report (default is 60).
<b>PDSE</b>	Is the data set a PDSE?
<b>REBLOCK REBLK</b>	Is the data set reblockable?
<b>RECFM RECFMT</b>	Prints the record format of the data set.
<b>SECALLOC ALLOCSEC</b>	Prints the secondary allocation amount for the data set.



Parameters	Description
<b>STORCLAS SC</b>	Prints the storage class for the data set, if there is one.
<b>TITLE=xxx</b>	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
<b>TOTALS</b>	Prints totals for the data set space allocations. If this parameter is not specified, totals does not print.
<b>UNIT ALLOCUNIT</b>	Prints the allocation unit (TRKS, CYLS, and for forth) for the data set.
<b>USERRED%</b>	% user data reduction
<b>VOLSER</b>	Prints the data set's volume serial number.

## Sample JCL (ACBJBAOD) 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 ISPF FILE, WHERE THE GENERATED REPORT IS SAVED
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY
//* DATA SET.
//*
//*****
//ALCISPFL EXEC PGM=IEFBR14
//ISPF DD DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//*****
//*
//* DATA SET REPORT GENERATION STEP
//*
//* PARAMETER FOLLOWING ACBQBAR1 - SAVED ISMF LIST (INPUT)
//* ISPF - 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 | CFSIZE -> 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.
//* 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.        *
//*      LSETNAME       |              -> CF Lock set name.      @A1A*
//*      LSTRNAME       |              -> CF Lock Structure Name. @A1A*
/*****
//GENREP EXEC ACBJBAOF,
//      PLIB1=SYS1.DGTPLIB,
//      TABL2=userid.TEST.ISPTABL
//ISPFIL DD DSN=IBMUSER.DATASET.REPORT,DISP=OLD
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACQBAR1 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=*

```

## 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. You can sort on a field.

See [“Sample JCL \(ACBJBAOF\) for generating a report from ISMF-saved DASD volume list” on page 377](#) 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
<b>%FREE</b>	Prints the % of free space on the volume.
<b>ALLOCSPC   ALLOCSP</b>	Prints the amount of allocated space on the volume.
<b>ALSPCTRK   ALSPTRK</b>	Prints the track-managed allocated space on the volume.
<b>CACHEFW</b>	Does the volume have Cache Fast Write enabled for it?

Parameters	Description
<b>CFVOLST</b>	CF Volume Status
<b>DASDFW</b>	Does the volume have DASD FAST WRITE enabled for it?
<b>DEVICE   DEVICETYPE</b>	Prints the device type of the volume.
<b>DEVNUM   ADDRESS</b>	Prints the device number of the volume.
<b>DUPLEX   DUPLEXST</b>	Is the volume part of a dual-copy pair?
<b>FRAG   FRAGINDX</b>	Prints the fragmentation index of the Volume.
<b>FREEDSCB   DSCBFREE</b>	Prints the number of free data set control blocks in the VTOC of the volume.
<b>FREESPC   FREESP</b>	Prints the volume's free space in KB.
<b>FREEVIRS   VIRSFREE</b>	Prints the number of free VIRs in the indexed VTOC on the volume.
<b>FREEEXT   EXTFREE</b>	Prints the number of free extents on the Vvolume.
<b>FRSPCTRK   FRSPTRK</b>	Prints the track-managed free space on the volume.
<b>INDXSTAT   STATINDX</b>	Prints the status of the indexed VTOC on the volume.
<b>INDEX%USED</b>	Prints the % of Index used on the volume.
<b>INIASRES</b>	Prints if volume is initialized as reserved.
<b>LGEXTRK   EXTLGTRK</b>	Prints the size of the largest track-managed extent available on the volume.
<b>LRGEXT   EXTLRG</b>	Prints the size of the largest extent available on the volume.
<b>OTHER   OTHERDEV</b>	If duplex, what is the secondary volume?
<b>OWNERID</b>	Prints the volume's owner ID.
<b>PAGELength=nn</b>	Sets page length for the report (default is 60).
<b>PHYSTAT   STATPHYS</b>	Prints the physical status of the volume.
<b>RDCACHE   RDSTAT</b>	Does the volume have READ CACHE enabled for it?
<b>SHARE   SHRDASD</b>	Is the DASD volume genned as SHARED?
<b>STORGRP   SG</b>	Prints the storage group that the volume belongs to.
<b>SUBSYS   SUBSYSID</b>	Prints the controller's subsystem identifier.
<b>TITLE=xxx</b>	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
<b>TOTALS</b>	Prints totals for the volume. If this parameter is not specified, totals does not print.
<b>USE   USEATTR</b>	Prints the volumes use attribute (public and so on).
<b>VOLSER</b>	Prints the volume serial number.
<b>VTOC%USED</b>	Prints the % of VTOC used on the volume.

## Sample JCL (ACBJBAOF) for generating a report from ISMF-saved DASD volume list

```

//*****
//*
//* SAMPLE JCL TO SORT A PREVIOUSLY SAVED DASD VOLUME LIST AND @D1C*
//* GENERATE A DASD VOLUME REPORT FROM IT. @D1C*
//*
//* 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 ISPFIL, WHERE THE GENERATED REPORT IS SAVED
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY
//* DATA SET.
//*
//*****
//ALCISPFL EXEC PGM=IEFBR14
//ISPFIL 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 *
/*
//*****
//*
//* STEP TO SORT A PREVIOUSLY SAVED DASD VOLUME LIST AND @D1C*
//* GENERATE A DASD VOLUME REPORT FROM IT. @D1A*
//*
//* PARAMETERS FOLLOWING ACBQVAR1 - ISMF SAVED LIST (INPUT); @D1C*
//* SORT(column name) (INPUT) - 3@D1A*
//* to sort list by the data column *
//* in ascending order. *
//* ISPFIL - 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 *
//* FRSPCTRK | FRSPTRK -> Free Space in TRK-Managed @A0A*
//* %FREE -> % Free *
//* ALLOCSPC | ALLOCSP -> Allocated Space *
//* ALSPCTRK | ALSPTRK -> Alloc Space in TRK-Managed @A0A*
//* FRAG | FRAGINDX -> Fragmentation Index *
//* LRTEXT | EXTLRG -> Largest Extent *
//* LGEXTRK | EXTLGTRK -> Largest Ext in TRK-Managed @A0A*
//* FREEEXT | EXTFREE -> Free Extents *
//* INDXSTAT | STATINDX -> Index Status *
//* FREEDSCB | DSCBFREE -> Free DSCBS *
//* FREEVIRS | VIRSFREE -> Free Vtoc Index Records *
//* DEVICE | DEVICETYPE -> Device Type *
//* DEVDNUM | 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 *
//* VTOC%USED -> VTOC % USED *
//* INDEX%USED -> INDEX % USED *
//* INIASRES -> Initialized as Reserved *
//* OWNERID -> Owner ID *
//*****
//GENREP EXEC ACBJBAOB,
// PLIB1=SYS1.DGTPLIB,
// TABL2=userid.TEST.ISPTABL
//ISPFIL DD DSN=IBMUSER.DASDVOL.REPORT,DISP=OLD
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQVAR1 DASDLIST SORT(column name)) +
NEWAPPL(DGT) 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=IBUSER.DASDVOL.REPORT,DISP=SHR
//SYSUT2   DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
//*

```

## 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 [“Sample JCL \(ACBJBAOT\) for generating a report from ISMF-saved tape list” on page 380](#) 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
<b>CKPTVOL   CKPT</b>	Prints whether this is a checkpoint volume.
<b>COMPTYPE   TYPECOMP</b>	Prints the compaction type.
<b>CRTDTVOL   VOLCRTDT</b>	Prints the volume's creation date.
<b>EXPDTVOL   VOLEXPDT</b>	Prints the volume's expiration date.
<b>LSTENTDT   LASTEJECT</b>	Prints the last eject/entry date.
<b>LSTMNTDT   LASTMOUNT</b>	Prints the volume's last mount date.
<b>LSTWRDT   LASTWRITE</b>	Prints the volume's last write date.
<b>LIBNAME</b>	Prints the volume's library name.
<b>MEDIA   MEDIATYPE</b>	Prints the volume's media type.
<b>OWNER   OWNERINFO</b>	Prints the volume's owner information.
<b>PAGELength=nn</b>	Sets the page length for the report (default is 60)
<b>RECTECH   TECHREC</b>	Prints the volume's recording technology.
<b>SHELF   SHELFLOC</b>	Prints the volume's shelf location.
<b>SPCLATTR   ATTRSPCL</b>	Prints the volume's special attributes.
<b>STORGRP   SGNAME</b>	Prints the storage group that the tape belongs to.
<b>TITLE=nn</b>	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks and cannot expand more than one record in length.
<b>USEATTR   ATTRUSE</b>	Prints the volume use attributes.
<b>VOLERROR   ERRORVOL</b>	Prints the volume's error information.
<b>VOLLOC   LOCVOL</b>	Prints the volume's location.
<b>VOLSER</b>	Prints the volume serial number.

Parameters	Description
WRTPROT   PROTWRT	Is the volume write protected?

## Sample JCL (ACBJBAOT) 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 ISPFIL, WHERE THE GENERATED REPORT IS SAVED
//* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY
//* DATA SET.
//*
//*****
//ALCISPFL EXEC PGM=IEFBR14
//ISPFIL DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),
//      BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//*****
//*
//* REPORT GENERATION STEP
//*
//*     PARAMETER FOLLOWING ACBQBAR4 - ISMF SAVED LIST NAME (INPUT)
//*     ISPFIL - 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
//ISPFIL 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
/*
//*****
//*
//* 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=*
//*****

```

## 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 [“Sample JCL \(ACBJBARD\) for generating data set report from DCOLLECT data”](#) on page 382 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
<b>ALLOCSP   ALLOCSPC</b>	Prints the data sets allocated space
<b>BACKUP   BACKUPDT</b>	Prints the date of the last backup for data set.
<b>BLKUNUSED</b>	Prints the number of unused blocks for the data set.
<b>CHANGE   CHGIND</b>	Is the change indicator on for data set?
<b>CREATE   CREATEDT</b>	Prints the data set's creation date.
<b>DATACLAS   DC</b>	Prints the data class for the data set, if there is one.
<b>DSNAME   DSN</b>	Prints the data set name.
<b>DSNLENGTH=nn</b>	Limits print of data set name to nn characters. Defaults to 44 characters.
<b>DSORG</b>	Prints the DSORG for data set.
<b>ENTRYTYPE</b>	Prints the data set's entry type.
<b>EXPIRE   EXPIREDT</b>	Prints the data set's expiration date.
<b>LASTREF   LASTREFDT</b>	Prints the data set's last reference date.
<b>LRECL</b>	Prints the data set's record length.
<b>MGMTCLS</b>	Prints the management class for the data set, if there is one.
<b>MULTVOL</b>	Is the data set multivolume?
<b>NUMEXT   EXTNUM</b>	Prints the number of extents for data set.
<b>PAGELength=nn</b>	Sets the page length for the report (default is 60).
<b>PDSE</b>	Is the data set a PDSE?
<b>REBLOCK   REBLK</b>	Is the data set reblockable?
<b>RECFM   RECFMT</b>	Prints the data set's record format.
<b>SMS   MANAGED</b>	Is the data set SMS-managed?
<b>STORCLAS   SC</b>	Prints the storage class for data set, if there is one.
<b>STORGRP   SG</b>	Prints the storage group for data set, if there is one.
<b>TITLE=xxx</b>	Prints a title for the report. Title does <i>not</i> need to be placed in parentheses or quotation marks, and cannot expand more than one record in length.
<b>TOTALS</b>	Specifies whether you want DSN space totals printed for this DCOLLECT data.
<b>USED%</b>	Prints the percentage of used space for the data set.
<b>VOLSEQ</b>	Prints the volume sequence number for data set.
<b>VOLSER</b>	Prints the volume serial of the data set.
<b>VVRCHK   VVR</b>	If the data set is SMS-managed, does it have a valid VVR or NVR?

## Sample JCL (ACBJBARD) 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 ISPFIL, WHERE THE GENERATED REPORT IS SAVED
//NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY
//DATA SET.
//
/*****
//ALCISPFL EXEC PGM=IEFBR14
//ISPFIL DD DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
//BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
/*****
//
//REPORT GENERATION STEP
//
//DCOLIN - DCOLLECT DATA (INPUT)
//ISPFIL - 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:
//
//ALLOCS | 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
//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 MULTI-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
//ISPFIL 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
ALLOCS
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=*
//*****

```

## 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 70 on page 384](#) 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 *  
//* *  
//*   MODCMDSPC - SPACE ATTRIBUTES OF MODEL COMMANDS DATA SET *  
//*   (optional) WHERE THE GENERATED OUTPUT WILL BE PLACED. *  
//*               THIS IS A SEQUENTIAL DATA SET WITH THE NAME *  
//*               <userid>.M<nn>.MODEL CMD, WHERE <userid> IS *  
//*               THE USER ID, <nn> IS A SERIAL NUMBER CHOSEN BY *  
//*               NAVIQUEST. *  
//* *  
//*               SYNTAX ==> MODCMDSPC(PRI,SEC) WHERE *  
//*               PRI -> PRIMARY SPACE QUANTITY IN *  
//*               CYLINDERS (1 TO 16777215), *  
//*               SEC -> SECONDARY SPACE QUANTITY IN *  
//*               CYLINDERS (0 TO 16777215). *  
//*               DEFAULTS ==> PRIMARY SPACE QUANTITY: 15 TRACKS, *  
//*               SECONDARY SPACE QUANTITY: 30 TRACKS. *  
//* *  
//*   REST OF THE PARAMETERS - MODEL COMMAND FOR GENERATION *  
//* *  
//*****  
//GENCMDS EXEC ACBJBA0B,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
//PROFILE PREFIX(IBMUSER)  
//ISPSTART CMD(%ACBQBAM1 DSNLIST +  
//MODCMDSPC(1,2) +  
//DD DSN=/,VOL=SER=@ +  
//) +  
//BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
// *  
//*
```

Figure 70. Sample JCL for ACBJBAM1

## Creating SMS 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 335](#), 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.

For more information on generating saved lists from ISMF tables, refer to [\*z/OS DFSMS Using the Interactive Storage Management Facility\*](#).

## 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). 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.

- 
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.

- 
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.

- 
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.

**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.

## 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

## 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.

- 
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.

- 
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,

**Y (yes)**

If you want to allocate the model commands data set with your own space attributes. This data set will contain the generated model commands.

**N (no)**

If you want to use NaviQuest default space attributes for the model commands data set. NaviQuest will allocate a cataloged sequential data set with primary space quantity of 15 tracks and secondary space quantity of 30 tracks.

- In the third field,

ALTER / MANAGEMENTCLAS(newmgmtclas)

*Example of Model Command:*

```
ALTER / MGMTCLAS(MCSTAND)
```

- 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)

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.Mxx.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 previously shown.

---

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=ADDRSSU, PARM='TRACE=YES'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFRAG DYNAM(SCR001)
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 SMS 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

This section documents intended Programming Interfaces that allow you to write programs to obtain the services of SMS.

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.

## 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
-----
                        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 71. Copy FILTLISTS Entry Panel*

## Appendix A. Sample batch job for CICS definitions

In order to utilize the DFSMSdss facilities to process data sets that are open to CICS TS files, a number of definitions and parameters in CICS TS need to be defined. The following sample batch job defines these to CICS.

When a storage administrator indicates with SMS policies that CICS file system interfaces should be called if DFSMSdss cannot obtain serialization to process the request, DFSMSdss will attempt to use the CICS file system interfaces to first close the files open to data sets in CICS and then to process the DFSMSdss request. Subsequently, these interfaces will be called to allow these files to be opened by CICS.

```
//DEFCSD JOB ,JBCAMMA,MSGCLASS=H,CLASS=A,
// MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID,TIME=1449
//*****
//* CONFIGURING THE CICS TS FOR DFSMSDSS TO CLOSE AND OPEN FILES:
//*
//* YOU NEED TO SET A NUMBER OF DEFINITIONS AND PARAMETERS IN
//* CICS TO ENABLE THE DFSMSDSS CICS SERVER CODE TO PROCESS
//* REQUESTS TO CLOSE OR OPEN FILES.
//*
//* TO ADD THE CICS RESOURCE DEFINITIONS THAT ARE REQUIRED
//* BY DFSMSDSS YOU NEED TO RUN THE RDO COMMANDS IN THE EXAMPLE BELOW.
//*
//* THIS EXAMPLE CREATES TWO RESOURCE GROUPS, DSSCICS AND DSSEXCI.
//* BEFORE RUNNING THE JOB YOU NEED TO EDIT IT TO CHANGE THE DEFAULT
//* LIBRARY VALUES TO MATCH THOSE OF YOUR INSTALLATION:
//*
//* + GROUP DSSCICS CONTAINS THE DEFINITIONS FOR THE DFSMSDSS CICS
//* SERVER PROGRAM. YOU NEED TO ADD THIS GROUP TO THE GROUP LIST
//* THAT IS USED AT CICS STARTUP.
//*
//* + DSSEXCI CONTAINS DEFINITIONS FOR A EXTERNAL CICS INTERFACE
//* (EXCI) GENERIC CONNECTION BECAUSE DFSMSDSS USES THE EXCI.
//* YOU NEED TO DEFINE THE DSSEXCI GROUP IF YOU DO NOT ALREADY
//* HAVE A EXCI GENERIC CONNECTION DEFINED.
//*****
//CONSDDEF EXEC PGM=DFHCSDUP
//STEPLIB DD DSN=CICSTS.V4R1.PROD.CICS.SDFHLOAD,DISP=SHR
//DFHCSD DD DSN=USRLCL.V4R1.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
DELETE LIST(MYLIST)
DELETE GROUP(DSSTEST)
DELETE GROUP(DSSCICS)
DELETE GROUP(DSSEXCI)

APPEND LIST(DFHLIST) TO(MYLIST)
ADD GROUP(DSSCICS) LIST(MYLIST)
ADD GROUP(DSSEXCI) LIST(MYLIST)

DELETE TERM(MAST) GROUP(DSSCICS)
DEFINE TERM(MAST) GROUP(DSSCICS) TYPETERM(DFHCONS)
    CONSNAME(MASTER1) DESCRIPTION(MVS CONSOLE CONS500)

DELETE TERM(TST1) GROUP(DSSCICS)
DEFINE TERM(TST1) GROUP(DSSCICS) TYPETERM(DFHCONS)
    CONSNAME(LOCALC02) DESCRIPTION(MVS CONSOLE LOCALC02)

DELETE TRANSACTION(DSSX) GROUP(DSSCICS)
DEFINE TRANSACTION(DSSX) GROUP(DSSCICS) PROGRAM(DFHMIRS)
    DESCRIPTION(DSS SERVER TRANSACTION REQUIRED BY DFSMSDSS)
    PROFILE(DFHCICSA)

DELETE PROGRAM(ADREXCIS) GROUP(DSSCICS)
DEFINE PROGRAM(ADREXCIS) GROUP(DSSCICS) LANGUAGE(ASSEMBLER)
    DESCRIPTION(CICS SERVER PROGRAM REQUIRED BY DFSMSDSS)
    DATALOCATION(ANY) EXECKEY(USER)

DELETE CONNECTION(DSSG) GROUP(DSSEXCI)
DEFINE CONNECTION(DSSG) GROUP(DSSEXCI)
    DESCRIPTION(EXCI GENERIC CONNECTION REQUIRED BY DFSMSDSS)
    ACCESSMETHOD(IRC) SINGLESESS(NO)
    PROTOCOL(EXCI) CONNTYPE(GENERIC)
```

```

    DATASTREAM(USER) RECORDFORMAT(U) AUTOCONNECT(NO)
    INSERVICE(YES) ATTACHSEC(IDENTIFY)

DELETE SESSIONS(DSSG) GROUP(DSEXCI)
DEFINE SESSIONS(DSSG) GROUP(DSEXCI)
    DESCRIPTION(EXCI GENERIC SESSIONS DEFINITION REQUIRED BY DFSMSDSS)
    CONNECTION(DSSG) PROTOCOL(EXCI) MAXIMUM(0,0)
    RECEIVEPFX(RG) RECEIVECOUNT(10)
    SENDSIZE(4096) RECEIVESIZE(4096)
    SESSPRIORITY(0) AUTOCONNECT(NO)
    BUILDCHAIN(YES) IOAREALEN(4096,4096) RELREQ(NO)
    DISCREQ(NO) NEPCCLASS(0)
    RECOVOPTION(SYSDEFAULT)

LIST LIST(MYLIST) OBJECTS
/*

```

---

## Appendix B. Accessibility

Accessible publications for this product are offered through [IBM Documentation for z/OS \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see [How to Send Feedback to IBM](#) to leave documentation feedback.



## Notices

---

This information was developed for products and services that are offered in the USA or elsewhere.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan*

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

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

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation  
Site Counsel  
2455 South Road*

Poughkeepsie, NY 12601-5400  
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document 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.

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

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

#### COPYRIGHT LICENSE:

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

## Terms and conditions for product documentation

---

Permissions for the use of these publications are granted subject to the following terms and conditions.

### **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

### **Personal use**

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

### **Commercial use**

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or



reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## IBM Online Privacy Statement

---

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at [ibm.com/privacy](http://ibm.com/privacy) and IBM's Online Privacy Statement at [ibm.com/privacy/details](http://ibm.com/privacy/details) in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at [ibm.com/software/info/product-privacy](http://ibm.com/software/info/product-privacy).

## Policy for unsupported hardware

---

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

## Minimum supported hardware

---

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

## Programming interface information

---

This publication is intended to help you define, initialize, and maintain SMS and manage storage with the IBM Interactive Storage Management Facility (ISMF).

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 by an introductory statement to a chapter or section.

## Trademarks

---

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

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

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

---

# Index

## Special Characters

&ACCT\_JOB variable [271](#), [287](#)  
&ACCT\_STEP variable [271](#), [287](#)  
&ACSENVIR variable [162](#), [271](#), [282](#)  
&ALLVOL variable [272](#), [286](#)  
&ANYVOL variable [272](#), [286](#)  
&APPLIC variable [273](#)  
&BLKSIZE variable [273](#)  
&DB2SSID variable [273](#)  
&DD variable [273](#)  
&DEF\_DATACLAS variable [273](#)  
&DEF\_MGMTCLAS variable [274](#)  
&DEF\_STORCLAS variable [274](#)  
&DSN variable [274](#), [286](#)  
&DSNTYPE variable [274](#), [281](#)  
&DSORG variable [275](#)  
&DSOWNER variable [275](#)  
&DSTYPE variable [275](#)  
&EATTR variable [275](#)  
&EXPDT variable [276](#)  
&FILENUM variable [276](#)  
&GROUP variable [276](#)  
&HLQ variable [276](#)  
&JOB variable [276](#)  
&LABEL variable [276](#)  
&LIBNAME variable [276](#)  
&LLQ variable [277](#)  
&MAXSIZE [35](#)  
&MAXSIZE variable [277](#), [281](#)  
&MEMHLQ variable [277](#), [282](#)  
&MEMLLQ variable [277](#), [282](#)  
&MEMN variable [277](#), [282](#), [286](#)  
&MEMNQUAL variable [277](#), [282](#)  
&MSPARM variable [168](#), [278](#), [288](#)  
&MSPDEST variable [168](#), [277](#)  
&MSPOLICY variable [168](#), [278](#)  
&MSPOOL variable [168](#), [278](#)  
&NQUAL variable [278](#)  
&NVOL variable [278](#)  
&PGM variable [278](#)  
&RECORG variable [278](#)  
&RETPD variable [279](#)  
&SECLABL variable [279](#)  
&SIZE [35](#)  
&SIZE variable [279](#), [281](#)  
&SYSNAME variable [283](#)  
&SYSPLEX variable [283](#)  
&UNIT variable [280](#), [281](#)  
&USER variable [280](#)  
&USER\_ACSVAR variable [281](#), [286](#), [288](#)  
&XMODE variable [281](#)  
% freespace [130](#)

## Numerics

32-name support [14](#)

## A

ABACKUP [303](#)  
Abbackup copy technique [81](#)  
ABARS  
    processing  
        pre-DFSMS management classes [202](#)  
ACBJBAOB JCL [340](#)  
accessibility  
    contact IBM [395](#)  
accessing  
    DASD volume [43](#)  
    optical volume [50](#)  
    pool storage group [40](#)  
    remote authorization [312](#)  
    VIO storage group [40](#)  
ACDS  
    allocating [15](#)  
    definition [12](#)  
    JCL allocation [15](#)  
    recovery [213](#)  
    relationship with SCDS [12](#)  
    saving [14](#)  
    share options [16](#)  
    size calculation [13](#)  
ACS  
    class determination [172](#)  
    data set rename [169](#)  
    designing and testing ACS routines [149](#)  
    DFSMS NaviQuest [149](#)  
    environment  
        RMMPOOL [170](#)  
        RMMVRS [170](#)  
    language  
        boolean expressions [285](#)  
        comparison operators [284](#)  
        description [149](#)  
        special functions [286](#)  
        statements [289](#)  
        writing routines [267](#)  
    object  
        deleting from SCDS [173](#)  
        information [172](#)  
    read-only variables [233](#)  
    read-write variables [269](#)  
    routine  
        activating [335](#)  
        application selection, writing [151](#)  
        constants [267](#)  
        creating [150](#)  
        Distributed FileManager/MVS [100](#), [288](#)  
        dummy and SUBSYS data sets [150](#)  
        environments [162](#)  
        exit [168](#)  
        invoking ISPF/PDF editor [150](#)  
        language reference [267](#)  
        masks [267](#)

- ACS (*continued*)
  - routine (*continued*)
    - operations [149](#)
    - output listing [152](#)
    - SCDS validation [155](#)
    - storage class example [298](#)
    - SYSIN and SYSOUT data sets [150](#)
    - testing [157](#)
    - translating [151](#)
    - translation listing [152](#)
    - translation results [151](#)
    - validating [153](#)
    - validation results [155](#)
  - storage group determination [172](#)
  - variables
    - read-only [270](#)
    - read-write [269](#)
  - VSAM RLS [238](#)
- ACS routines
  - defining [149](#)
  - description [4](#)
  - security labels [173](#)
  - SMS class [5](#)
  - storage group [5](#), [176](#)
  - test cases [1](#)
  - testing [1](#), [5](#)
  - translating [1](#)
  - z/OS Network File System [150](#)
- ACS Test Case Define/Alter panel [178](#)
- ACTIVATE command
  - ISMF [300](#)
- ACTIVATE, ISMF command [180](#)
- activation
  - individual ACS routine [335](#)
- adding volumes to a storage group [42](#)
- additional volume amount [121](#), [206](#)
- address space
  - recovery [215](#)
  - starting [17](#)
- Admin or User command Backup [77](#)
- aggregate group
  - altering [207](#)
  - application [138](#)
  - attributes
    - altering [207](#)
    - defining [139](#)
    - editing [139](#), [207](#)
  - backup [140](#)
  - backup attributes [138](#)
  - copying SMS [208](#)
  - defining [137](#), [138](#), [141](#)
  - definition [4](#)
  - displaying list [202](#)
  - instruction data set [138](#)
  - names [139](#), [207](#)
  - panels [138](#)
  - planning [138](#)
  - recovering [140](#)
  - sort criteria [202](#)
  - view criteria [202](#)
- aggregate group application
  - understanding [137](#)
- alias device type [193](#)

- allocation
  - ACDS [15](#)
  - COMMDS [16](#)
  - control data set [11](#)
  - data set [101](#)
  - extended addressability [122](#)
  - retrying [126](#), [128](#)
  - SCDS [15](#)
  - spreading across multiple volumes [109](#)
  - storage class, specific volume [97](#)
  - threshold [38](#)
- allocation/migration threshold [38](#)
- ALLOCxx, member
  - VIO data sets [35](#)
- ALTER command
  - ISMF [300](#)
- ALTER(YES) keyword [255](#)
- altering
  - aggregate group [207](#)
  - copy pool [206](#)
  - data class [205](#)
  - management class [204](#)
  - storage
    - class [205](#)
    - group [203](#)
- ALX parameter [35](#)
- array DASD [92](#)
- assigning
  - data class [135](#)
  - management class [81](#)
  - storage class [100](#)
- assigning volumes
  - reserve storage pools [55](#)
- assistive technologies [395](#)
- attributes
  - aggregate group [138](#), [139](#), [207](#)
  - copy pool [144](#)
  - data class
    - record [118](#)
    - space [118](#)
    - VSAM [126](#)
  - GDG [76](#)
  - management class
    - backup [77](#)
    - expiration [70](#)
    - migration [73](#)
  - object class
    - backup [80](#)
    - transition [79](#)
  - point-in-time copy [93](#)
  - pool storage group [35](#)
  - retention period [72](#)
  - storage class
    - accessibility [93](#)
    - availability [85](#), [92](#)
    - defining for CF cache sets [98](#), [251](#)
    - defining use of zHyperLinks [99](#)
    - definition [84](#)
    - storage group [33](#), [34](#)
    - sustained data rate [91](#)
    - timing [27](#)
    - VSAM [128](#)
- attributes (data class)
  - data class

attributes (data class) *(continued)*

data class *(continued)*

VSAM [121](#)

AUDIT command

ISMF [300](#)

AUDIT parameter, bypass RACF logging [226](#)

auto backup

storage group field [36](#)

auto dump [36](#)

auto migrate [35](#)

AUTOALTER(YES) keyword [256](#)

Automated Tape Library Dataserver (ATLDS) [29](#)

automated testing, overview [327](#)

automatic class selection (ACS) [4](#)

automatic interval migration [35](#)

availability [85](#), [92](#)

availability, VSAM RLS [238](#)

Avgreg [120](#)

## B

backup

auto backup [36](#)

defining aggregate groups [140](#)

management class [77](#)

object copies [48](#)

backup control data set (BCDS) [315](#)

backup copy technique [78](#)

backup devices [93](#)

backup frequency [77](#)

backup-while-open (BWO) [133](#)

base configuration

active control data set [3](#)

adding a system [22](#)

altering [1](#)

CF cache structures [250](#)

completing [24](#)

control data set application selection [23](#), [24](#), [250](#)

coupling facility definition [23](#), [24](#), [250](#)

defining [1](#), [19](#)

deleting a system [22](#)

description [3](#)

displaying [1](#)

planning [19](#)

renaming a system [24](#)

SMS complex [3](#), [5](#), [22](#)

testing [1](#)

updating [1](#)

VSAM RLS [23](#), [24](#), [250](#)

base line test set [324](#)

BCDS [315](#)

bias [89](#)

BLKSZLIM keyword [130](#)

block size limit [130](#)

boolean expressions, ACS routine [285](#)

BOTTOM command

ISMF [300](#)

browsing, ACS routine translation [151](#), [153](#)

buffering, system-managed [123](#)

## C

CA reclaim parameter [133](#)

cache

device [88](#)

statistics [189](#)

cache fast write function [310](#)

cache set [23](#)

CANCEL command

ISMF [300](#)

capabilities, filtering [325](#)

capacity planning information [313](#)

CBROAMxx member [28](#), [34](#), [67](#)

CF cache structure

recovering [261](#)

CF cache structures

sharing [242](#)

CFRM (coupling facility resource manager) policy [238](#)

channel-to-channel adapter device number [141](#)

CICS [133](#), [235](#), [242](#), [245](#), [262](#)

CIsze data [129](#)

class transition attributes [67](#)

classification, data [326](#)

CLEAR command

ISMF [300](#)

cloud network connections

planning [57](#)

understanding [57](#)

clouds

defining [57](#), [58](#)

defining additional [60](#)

defining DS800 as an object proxy [58](#)

digital certificates

using DS8000 as an object proxy [60](#)

setting up digital certificates [59](#)

collecting, data [325](#)

collection

assigning [30](#)

definition [29](#)

multiple [30](#)

name

OAM [29](#)

qualifiers [278](#)

OAM object [51](#)

commands

ISMF commands

ACTIVATE [300](#)

ALTER [300](#)

AUDIT [300](#)

BOTTOM [300](#)

CANCEL [300](#)

CLEAR [300](#)

COMPRESS [300](#)

COPY [301](#)

DO [301](#)

DOWN [301](#)

DSUTIL [301](#)

DUMP [301](#)

END [301](#)

ERTB [301](#)

FILTER [301](#)

FIND [301](#)

FOLD [301](#)

HELP [301](#)

LIBRARY [301](#)

LISTPRT [301](#)

PROFILE [302](#)

- commands (*continued*)
  - ISMF commands (*continued*)
    - QRETRIEV [302](#)
    - QSAVE [302](#)
    - REFRESH [302](#)
    - RELEASE [302](#)
    - RESHOW [302](#)
    - RESTORE [302](#)
    - RETURN [302](#)
    - RIGHT [302](#)
    - SAVE [302](#)
    - SORT [302](#)
    - TOP [302](#)
    - UP [302](#)
    - VALIDATE [303](#)
    - VIEW [303](#)
- COMMDS
  - allocating [16](#)
  - current utilization statistics [12](#)
  - definition [12](#)
  - JCL allocation [16](#)
  - recovery [214](#)
  - size calculation [14](#)
  - sizing [13](#)
  - SMS complex [12](#)
- compaction [125](#)
- comparing regression testing [331](#)
- comparison operators, ACS routine [284](#)
- compatibility mode [20](#)
- COMPRESS command
  - ISMF [300](#)
- concurrent copy [30](#), [83](#), [84](#), [93](#)
- connection
  - storage group
    - to DASD volume [43](#)
    - to optical volume [50](#)
  - system to storage group [40](#), [237](#)
- constants, ACS routines [267](#)
- constructs
  - SMS configuration [4](#)
  - types [4](#)
- contact
  - z/OS [395](#)
- control data set
  - allocating [5](#), [11](#)
  - application selection panel [19](#), [23](#), [24](#)
  - background [5](#)
  - description [11](#)
  - fixed data fields [13](#)
  - ISMF primary option menu [2](#)
  - multivolume [15](#)
  - RACF [232](#)
  - recovery [213](#)
  - sharing [239](#)
  - sizing [13](#)
  - structure changes [11](#)
  - types
    - active control data set (ACDS) [11](#)
    - communications data set (COMMDS) [11](#)
    - source control data set (SCDS) [11](#)
- convert status for a DASD volume [26](#)
- converting volumes to SMS [212](#)
- CONVERTV command, DFSMSdss [161](#)
- CONVERTV command, dss [212](#)
- COPY command
  - ISMF [301](#)
- copy pool
  - altering [206](#)
  - application [144](#)
  - attributes [144](#)
  - defining [143](#), [144](#)
  - definition [4](#)
  - panels [144](#)
  - planning [143](#)
- copy pool backup storage group [38](#)
- copying SMS classes, storage and aggregate groups [208](#)
- COUPLExx member [22](#)
- coupling facility (CF)
  - cache and lock structures [238](#), [242](#)
  - cache structures [255](#)
  - defining cache structures [250](#)
  - displaying information [253](#)
  - lock structure [244](#), [255](#)
  - monitoring [253](#)
  - recovery [260](#)
  - requirements for VSAM RLS [235](#)
  - resource manager (CFRM) policy [238](#)
  - statistics monitoring [259](#)
  - system-managed duplexing [246](#)
  - system-managed lock structure duplexing [235](#), [244](#), [260](#)
  - varying cache structures [259](#)
- creating
  - customized tape reports [386](#)
  - data set name reports [385](#)
  - model commands [388](#)
  - SMS reports [384](#)
  - volume reports [385](#)
- cross-system coupling facility (XCF) [6](#)
- customizing ISMF [3](#)
- cycle end time [46](#), [49](#)
- cycle start time [46](#), [48](#)

## D

- DASD
  - fast write [88](#)
  - fast write function [310](#)
  - storage
    - control unit [309](#)
    - groups [25](#), [43](#)
    - object [27](#)
  - volume
    - access by storage group [43](#)
    - data set space [279](#)
    - deleting from the system [209](#)
    - status [26](#)
- DASD volume
  - initial status [26](#)
- DASDVOL alter authority [312](#)
- DASDVOL authority [44](#), [101](#)
- data class
  - altering [1](#), [205](#)
  - assigning [135](#)
  - attributes
    - description [118](#)
    - JCL processing [135](#)
    - Recorg [135](#)

- data class (*continued*)
  - creating additional [136](#)
  - defining
    - application selection [118](#)
    - record and space attributes [118](#)
    - volume [121](#), [126](#)
  - definition [117](#)
  - deleting [211](#)
  - description [4](#)
  - displaying [1](#)
  - list [201](#)
  - order when processing attributes [135](#)
  - planning [117](#)
  - record and space attributes [118](#)
  - specifying sort criteria [201](#)
  - understanding [117](#)
  - view [201](#)
  - volume attributes [121](#)
  - VSAM attributes [121](#), [126](#)
- data classification [324](#), [326](#)
- data collection [313](#)
- data DCOLLECT, input [325](#)
- data saved lists, input [325](#)
- data set
  - allocating space [21](#)
  - allocation [101](#), [280](#)
  - duplicate names [227](#)
  - empty [118](#)
  - extended addressability [122](#)
  - instruction [138](#)
  - mask [267](#)
  - name qualifiers [278](#)
  - name type [121](#)
  - nonrecoverable [239](#)
  - null [118](#)
  - open for VSAM RLS [252](#)
  - organization
    - DSORG value [128](#)
  - physical sequential (PS) [74](#), [91](#), [125](#), [129](#)
  - qualifiers [286](#)
  - read-only variables on rename [283](#)
  - recall processing [170](#)
  - RECORD [278](#)
  - recover processing [170](#)
  - recoverable [239](#)
  - renaming [283](#)
  - selection [138](#)
  - separation [21](#), [317](#)
- data set access
  - initial access response seconds [91](#)
- data set name reports, creating [385](#)
- data set placement
  - primary list [102](#)
  - secondary list [102](#)
  - tertiary list [102](#)
- data set profile, RACF [175](#)
- data set stacking
  - description [164](#)
  - read-write variables [270](#)
  - using
    - VOL=REF [165](#)
    - VOL=SER= [165](#)
- data SMF, input [325](#)
- data subtype [324](#)
- data testing, example [334](#)
- data type [324](#)
- data VMA, input [326](#)
- data-in-virtual (DIV)
  - size limit [14](#)
- data, collecting [325](#)
- Db2
  - object storage table [27](#)
- DCOLLECT command
  - active
    - data set [314](#)
    - volumes [314](#)
  - backup data [314](#)
  - capacity planning data [314](#)
  - description [313](#)
  - ISMF panels [313](#)
  - migration data [314](#)
  - pool storage groups [315](#)
  - storage groups [315](#)
  - volumes [315](#)
- default device geometry parameter [21](#)
- default management class
  - SCDS base define field [21](#)
  - specifying [67](#), [156](#)
- default management class parameter [21](#)
- default unit parameter [21](#)
- DEFINE CLUSTER command [95](#)
- defining CF cache and lock structures
  - sharing [242](#)
- defining clouds [57](#), [58](#)
- defining use of zHyperLinks [99](#)
- deleting
  - DASD volumes from the system [209](#)
  - SMS classes [211](#)
  - storage groups [210](#)
- destaging [311](#)
- determining errors [332](#)
- device information [309](#)
- device status [192](#)
- DEVSERV command
  - NOT-NPAV [198](#)
  - PATHS [199](#)
  - QDASD [199](#), [200](#)
  - QPAVS
    - NOT-ALIAS [197](#)
    - NOT-NPAV [195](#)
  - sample output [195](#)
  - solid state drives [200](#)
  - UNBOX [197](#)
- DEVSERV QPAVS command [110](#)
- DEVSUPxx member [130](#)
- DFSMS [1](#)
- DFSMSrmm
  - scratch tape pools [29](#)
  - storage groups [29](#)
  - tape storage groups [29](#)
  - Vital Record Specification (VRS) [69](#)
- diagram, tree [326](#)
- digital certificates for clouds
  - setting up [59](#)
- directory [120](#)
- DISALL
  - storage group to optical volume [51](#)



- DISALL (*continued*)
  - system to storage group [41](#)
- disaster recovery [93](#)
- disconnect sphere at CLOSE [99](#)
- discrete profile, RACF [175](#)
- DISNEW
  - storage group to optical volume [51](#)
- DISPLAY SMS command
  - cache statistics [189](#)
  - coupling facility (CF) information [253](#)
  - OAM status [185](#)
  - SMS maintenance [192](#), [193](#)
  - trace information [186](#)
- DISPLAY XCF command [253](#)
- Distributed FileManager/MVS
  - ACS routines
    - determining SMS classes [100](#), [150](#)
    - using ACS routine variables [288](#)
- DIV [14](#)
- DO command
  - ISMF [301](#)
- DO statement [292](#)
- DOWN command
  - ISMF [301](#)
- drive (optical)
  - displaying list [202](#)
  - sort criteria [202](#)
  - view criteria [202](#)
- drive start threshold [47](#), [49](#)
- DS separation profile parameter [20](#), [21](#)
- DSUTIL command
  - ISMF [301](#)
- dual copy [30](#), [83](#), [93](#)
- dump
  - auto dump [36](#)
  - classes [36](#)
  - dump system name [36](#)
- dump class [38](#)
- DUMP command
  - ISMF [301](#)
- duplexing [311](#)
- duplicate data set names [227](#)
- dynamic data set
  - cache management [88](#)
- dynamic volume count parameter [124](#)

## E

- edit a data set [139](#), [208](#)
- ENABLE
  - storage group to optical volume [50](#)
  - system to storage group [41](#)
- END command
  - ISMF [301](#)
- END statement
  - ACS routine
    - description [297](#)
- end user
  - ISMF primary option menu [2](#)
- ENF72 [38](#)
- enhancements, miscellaneous [387](#)
- Enterprise Storage Server (ESS)
  - FlashCopy [83](#)
  - SnapShot [83](#)

- error messages
  - SMS PAV option [113](#)
- errors
  - definition [324](#)
  - determining [332](#)
- ERTB command
  - ISMF [301](#)
- example
  - using DD PROTECT parameter in JCL [176](#)
  - using DD SECMODEL parameter in JCL [176](#)
- examples
  - data testing [334](#)
- EXIT statement [295](#)
- exit, pre-ACS routine [168](#)
- Expdt (data class field) [121](#)
- expected results [324](#)
- expected results, saving [333](#)
- expiration
  - EXPDT [70](#)
  - management class attributes [70](#)
  - object class
    - backup attributes [80](#)
    - transition attributes [79](#)
    - retention limit [70](#)
- extend storage group [37](#)
- extended addressability [122](#)
- extended format [128](#), [131](#)
- extended remote copy (XRC) [83](#)
- Extent Constraint Removal parameter [132](#)

## F

- FALLBACK command [263](#)
- fallback, VSAM RLS [262](#)
- false lock contention [246](#)
- FIELD resource class [232](#)
- file systems
  - disk sublevel 2 for [27](#)
- FILTER command
  - ISMF [301](#)
- FILTLIST statement [290](#)
- FIND command
  - ISMF [301](#)
- fixed data fields, sizing [13](#)
- FlashCopy [83](#), [84](#), [93](#), [143](#)
- FOLD command
  - ISMF [301](#)
- formula
  - CF lock structure [244](#), [247](#)
  - SHCDS size [240](#)
- FRlog parameter [134](#)

## G

- GDG
  - management attributes [76](#)
  - storage management [69](#)
  - using data set separation with [321](#)
- generating test cases [329](#)
- global connectivity, examples [237](#)
- global shared resources (GRS) [6](#)
- guaranteed backup frequency [40](#)
- guaranteed space [91](#), [92](#), [95](#), [97](#)



guaranteed synchronous write [98](#)

## H

HCD (hardware configuration definition) [34](#)

HELP command

ISMF [301](#)

## I

IART (initial access response time) [90](#)

ICKDSF (Device Support Facilities)

defining reserve storage pool [55](#)

IF statement [293](#)

IGDSMSxx

creating [17](#)

description [17](#)

modifying for VSAM RLS [248](#)

IGDSMSxx parmlib member

CF\_TIME parameter [248](#), [257](#)

DEADLOCK\_DETECTION parameter [248](#), [257](#)

DSNAME parameter [248](#)

RLS\_MAX\_POOL\_SIZE parameter [243](#), [248](#), [258](#)

RLS\_MaxCfFeatureLevel parameter [248](#), [257](#)

RLS\_MAXCfFeatureLevel parameter [132](#)

RLsAboveTheBarMaxPoolSize parameter [258](#)

RLsFixedPoolSize parameter [258](#)

RLSINIT parameter [248](#), [257](#)

SMF\_TIME parameter [248](#), [258](#)

STEPNAME parameter [248](#)

VOLSELMSG parameter [248](#)

IGDSSIIN

SMS initialization [179](#)

IGWLOCK00, CF lock structure [244](#)

IKJTSOxx member [310](#)

implementation phase [324](#)

INIT command [55](#)

initial access response seconds [90](#)

initial program load (IPL) [179](#)

initial test [324](#)

input data

saved lists test [325](#)

installation defaults [19](#)

instantaneous replication [93](#)

instruction data set

aggregate group field [208](#)

definition [138](#)

Interactive Storage Management Facility [3](#)

Interactive Storage Management Facility (ISMF) [1](#)

Interactive System Productivity Facility (ISPF)

authorizing access [233](#)

PDF editor [150](#)

interval migration, automatic [35](#)

IODF [193](#)

IPL [179](#)

ISMF

ACTIVATE command [180](#)

aggregate group application [137](#)

application module names [219](#)

applications [1](#)

automatic class selection application [149](#)

commands

ACTIVE [300](#)

ISMF (*continued*)

commands (*continued*)

ALTER [300](#)

AUDIT [300](#)

BOTTOM [300](#)

CANCEL [300](#)

CLEAR [300](#)

COMPRESS [300](#)

COPY [301](#)

DO [301](#)

DOWN [301](#)

DSUTIL [301](#)

DUMP [301](#)

END [301](#)

ERTB [301](#)

FILTER [301](#)

FIND [301](#)

FOLD [301](#)

HELP [301](#)

summary [299](#)

control data set [19](#)

copy pool application [143](#)

data class application [117](#)

description [1](#)

dialog [15](#)

displaying CF information [254](#)

end user [2](#)

function module names [219](#)

line operator [221](#)

line operator module names [221](#)

management class application [67](#)

module names

line operators [221](#)

locating [219](#)

primary option menu [2](#), [18](#)

RACF interactions [218](#)

storage administrator [2](#)

storage administrator functions [1](#)

storage group application [25](#), [32](#)

storage group status [189](#)

ISMF (Interactive Storage Management Facility)

customization [3](#)

restrictions to customizing [3](#)

ISMF primary option menu

access [5](#)

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 [5](#)

storage class [2](#)

storage group [2](#)

IXCL1DSU Couple Data Set format utility [244](#)

## J

JCL (job control language)

DD PROTECT parameter, example [176](#)

DD SECMODEL parameter, example [176](#)

extracting security labels [174](#)

JCL keyword

JCL keyword (*continued*)  
DISP=OLD [83](#)  
STORCLAS=DUPT@SMS [83](#)  
UNIT= [83](#)  
VOL=SER= [83](#)  
JES3 [23](#)

## K

key ranges [95](#)  
keyboard  
  navigation [395](#)  
  PF keys [395](#)  
  shortcut keys [395](#)  
Keylen [129](#)  
Keyoff [129](#)

## L

LEFT  
  commands  
    LEFT [301](#)  
    LIBRARY [301](#)  
    LISTPRT [301](#)  
    PROFILE [302](#)  
    QRETRIEV [302](#)  
    QSAVE [302](#)  
    REFRESH [302](#)  
    RELEASE [302](#)  
    RESHOW [302](#)  
    RESTORE [302](#)  
    RETURN [302](#)  
    RIGHT [302](#)  
    SAVE [302](#)  
    SORT [302](#)  
    TOP [302](#)  
    UP [302](#)  
    VALIDATE [303](#)  
    VIEW [303](#)  
LEFT command  
  ISMF [301](#)  
level 0 storage [73](#)  
library  
  displaying list [202](#)  
  names, defining [46](#), [49](#)  
  sort criteria [202](#)  
  view criteria [202](#)  
LIBRARY command  
  ISMF [301](#)  
library names [47](#), [49](#)  
library, test bed [324](#)  
line operators  
  summary [303–307](#)  
LIST command [200](#)  
listing  
  aggregate group [202](#)  
  data class [201](#)  
  management class [202](#)  
  optical  
    drive [202](#)  
    library [202](#)  
  storage  
    class [202](#)

listing (*continued*)  
  storage (*continued*)  
    group [202](#)  
    tape data class [202](#)  
    volumes in a storage group [53](#)  
LISTPRT command  
  ISMF [301](#)  
LISTSYS line operator [189](#)  
LISTVOL line operator  
  listing volumes  
    DASD [53](#)  
    mountable optical volume list [53](#)  
    object and object backup [53](#)  
  OAM status [185](#)  
  storage group  
    DASD [26](#)  
    tape [29](#)  
lock set [24](#)  
lock structure  
  sharing control [239](#)  
lock structure, CF  
  adjusting size [247](#)  
  defining [244](#)  
  false contention [246](#)  
  retained locks [245](#)  
LOG parameter  
  data class [134](#)  
  DEFINE CLUSTER command [252](#)  
log replicate [135](#)  
logical subsystem [93](#)  
logstream ID [134](#)  
LOGWRITE statement [297](#)  
LRECL [119](#)

## M

maintaining [185](#)  
management class  
  ACS routine [19](#)  
  altering [1](#), [204](#)  
  application selection [69](#), [72](#)  
  assigning [81](#)  
  attributes  
    backup [80](#)  
    defining [69](#), [70](#)  
    expiration [70](#)  
    GDG [76](#)  
    migration [73](#)  
    retention limit [72](#)  
  backup  
    attributes [77](#)  
    concurrent copy [78](#)  
    copy technique [78](#)  
    frequency [77](#)  
    version retention period [77](#)  
  default [19](#), [21](#), [67](#)  
  defining [1](#), [67](#)  
  defining base configuration information [19](#)  
  deleting [211](#)  
  describing [68](#)  
  description [4](#)  
  displaying [1](#)  
  displaying list [202](#)  
  expiration attributes [70](#)

- management class (*continued*)
  - [NOACTION 19](#)
  - [OAM 67](#)
  - [planning 68](#)
  - [RACF 232](#)
  - [retention limit 70](#)
  - [sort criteria 202](#)
  - [understanding 67](#)
  - [view criteria 202](#)
- management class name [139, 207](#)
- manual tape library (MTL) [29](#)
- masks, ACS language [267](#)
- MAXSYSTEM value [244](#)
- media interchange [130](#)
- media type [128, 130](#)
- member name [139, 208](#)
- menu, ISMF primary option [323](#)
- method considerations [326](#)
- MGMTCLAS syntax
  - JCL, TSO/E, IDCAMS parameter [81](#)
  - OSREQ STORE or CHANGE parameter [82](#)
- migrated storage [73](#)
- migration
  - auto migrate [35](#)
  - automatic interval [35](#)
  - data
    - collection [314](#)
    - information [313](#)
  - level 1 storage [73](#)
  - level 2 storage [73](#)
  - management class attributes [73](#)
  - on-demand [35](#)
  - pool storage group [38](#)
  - threshold [38](#)
- migration attributes [74](#)
- migration control data set (MCDS) [314](#)
- millisecond response [85](#)
- millisecond response for storage class [88](#)
- millisecond response time (MSR)
  - defining [86](#)
  - defining for solid state drives [88](#)
- miscellaneous enhancements [387](#)
- model commands, creating [388](#)
- modifying IGDSMSxx [248](#)
- module name, ISMF
  - commands [221](#)
  - functions [219](#)
  - line operators [221](#)
- monitoring
  - coupling facility (CF) [253](#)
  - statistics [259](#)
- monitoring for false lock contention [246](#)
- multi-tiered storage groups [98, 101, 109](#)
- multilevel security labels in ACS routines
  - creating a security label in the RACF profile [175](#)
  - extracting the security label from the RACF profile [176](#)
  - installation tasks [174](#)
  - overview [173](#)
  - planning tasks [174](#)
  - specifying the &SECLABL read-only variable [176](#)
  - writing ACS test cases using SECLABEL [178](#)
- multiple allegiance [98](#)
- MVS
  - command for displaying CF information [253](#)

- MXIG parameter [35](#)

## N

- naming
  - SHCDS [240](#)
- navigation
  - keyboard [395](#)
- NaviQuest
  - designing and testing ACS routines [149](#)
  - maintaining SMS configuration [185](#)
  - storage management tasks in batch [1](#)
- network connection
  - definition of [61](#)
- network connection construct
  - definition of [61](#)
- networks
  - connections for object storage networks [63](#)
  - defining additional networks [66](#)
  - defining connections [61](#)
  - defining connections to object storage clouds [63](#)
  - defining network connections attributes [63](#)
  - planning network connections [63](#)
  - securing network communication [61](#)
  - setting up digital certificates for cloud [61](#)
  - understanding [61](#)
- new terminology [323](#)
- NOTCON
  - storage group to optical volume [51](#)
  - system to storage group [41](#)
- number of backup versions [77](#)
- number of copies [139, 207](#)
- numeric read-only ACS variables [285](#)

## O

- OAM
  - activating SCDSs [184](#)
  - backup storage group [27](#)
  - cataloging [29](#)
  - class transition attributes [67](#)
  - collection names [29](#)
  - DISPLAY SMS command [185](#)
  - management class definition [67](#)
  - object collection [51](#)
  - OSMC [27](#)
  - restarting [184](#)
  - status [185, 189](#)
  - storage group
    - hierarchy [27](#)
    - object collection [31](#)
- OAMXCF parameter, DISPLAY SMS command [189](#)
- object
  - access method [185](#)
  - ACS information
    - deleting [173](#)
    - displaying [172](#)
  - backup storage group [27, 34, 50](#)
  - cataloging [29](#)
  - character strings [267](#)
  - class
    - ACS routines [267](#)
    - backup attributes [80](#)

- object (*continued*)
  - class (*continued*)
    - transition attributes [79](#)
  - collection [29, 30](#)
  - Db2 directory [29](#)
  - mask representation [267](#)
  - name [277](#)
  - reading [28](#)
  - status field [50](#)
  - storage group
    - access types [50](#)
    - ACS routines [267](#)
    - attributes [46, 48](#)
    - backup [6, 34](#)
    - defining [43](#)
    - definition [27](#)
    - devices [50](#)
    - hierarchy [27, 34](#)
    - qualifier [27](#)
    - SCDS [52](#)
    - timing attributes [27](#)
  - storage hierarchy [84](#)
  - write requests [49](#)
  - writing [28](#)
- object access
  - initial access response seconds [91](#)
- object storage hierarchy
  - disk sublevel 1 [27](#)
  - disk sublevel 2 [27](#)
  - optical volumes [27](#)
  - tape sublevel 1 [27](#)
  - tape sublevel 2 [27](#)
- on-demand migration [35](#)
- optical
  - library
    - names [46, 49](#)
    - pseudo [28](#)
    - sharing [28](#)
  - volume
    - access [50](#)
    - label [28](#)
    - table of contents [28](#)
- OSMC
  - storage management cycle [67](#)
- OSMC Processing System [47, 49](#)
- OSREQ STORE macro [70](#)
- output data set name prefix [139](#)
- overflow storage group [37](#)
- OVERWRITE SPACE [119](#)
- overview automated testing [327](#)

## P

- parallel access volume (PAV) option
  - defining new storage classes [111](#)
  - displaying information about [112](#)
  - error messages [113](#)
  - installation tasks [110](#)
  - modifying storage classes [111](#)
- parallel access volume option [109](#)
- parallel access volumes (PAV) [98](#)
- Parallel Sysplex
  - cross-system coupling facility (XCF) [6](#)
  - definition [6](#)

- Parallel Sysplex (*continued*)
  - displaying OAM status [189](#)
  - multiple SMS complexes [6](#)
  - MVS environment [6, 22](#)
  - VSAM RLS [238](#)
- partial release [73, 206](#)
- password protection [217](#)
- PAV
  - PAV-alias [193](#)
  - PAV-base [193](#)
- PDSE
  - sharing [6](#)
- peer-to-peer remote copy (PPRC) [83, 393](#)
- PERMIT command [175](#)
- phase test set [324](#)
- physical sequential data set [74, 91, 125, 129](#)
- planning
  - aggregate groups [138](#)
  - base configuration [19](#)
  - copy pools [143](#)
  - data class [117](#)
  - storage
    - class [84](#)
    - group [30](#)
- planning cloud network connections [57](#)
- point-in-time copy
  - concurrent copy [93](#)
  - FlashCopy [93](#)
  - SnapShot [93](#)
  - virtual concurrent copy [93](#)
- PPRC [83](#)
- pre-ACS interface
  - DFSMSrmm support [162](#)
- pre-ACS routine exit [168](#)
- prefix, subtype [324](#)
- primary
  - location [138](#)
  - storage [73](#)
- primary list
  - volume [105](#)
- primary option menu, ISMF [323](#)
- PROC statement [290](#)
- PROFILE command
  - ISMF [302](#)
- program control feature [218](#)
- programming interfaces
  - ACS routines [149, 267](#)
  - COPYFILT macro [390](#)
- PROTECT parameter [173, 176](#)
- Protecting NaviQuest [325](#)
- pseudo optical library [28](#)

## Q

- QDASD parameter [195](#)
- QRETRIEV command
  - ISMF [302](#)
- QSAVE command
  - ISMF [302](#)
- qualifier field [46](#)
- QUIALL, SMS status [41](#)

## R

### RACF

- authorization structure [219](#)
- bypass logging [226](#)
- DASDVOL alter authority [312](#)
- FACILITY profiles for storage administration [226](#)
- FIELD resource class [232](#)
- ISMF functions [218](#)
- program control feature [218](#)
- protecting modules [226](#)
- RDEFINE command [226](#)
- resource owner [218](#)

### RACF profile

- creating security labels [175](#)
- extracting security labels [176](#)

### RACF security

#### RALTER SECDATA command [175](#)

#### RAMAC 90

#### RAMAC Virtual Array [15](#)

#### RAUTH line operator [312](#)

#### RDEFINE command [226](#)

#### RDEFINE SECDATA command [175](#)

#### RDEFINE SECLABEL command [175](#)

#### read caching [310](#)

- read-only ACS variable, SECLABEL  
specifying in the ACS routine [176](#)

- read-only ACS variable, SECLABL  
overview [173](#)

#### read-only variables

- ACS routine [284](#)
- initializing [281](#)
- OAM [282](#)
- pre-ACS routine exit  
[168](#)
- restrictions [283](#)

#### reason code [195](#)

#### recall processing [170](#)

#### Recfm record format [119](#)

#### recommendations

- defining pool storage groups [26](#)
- defining VSAM RLS attributes [252](#)
- determining hardware requirements for VSAM RLS [235](#)
- disabling SMSVSAM [262](#)
- dynamic data set cache management [89](#)
- information recovery [213](#)
- protecting ISMF functions [218](#)
- STGADMIN.DPDSRN [227](#)
- using cache/DFW [89](#)
- using CF cache structures in JES3 environment [23](#)
- using control data sets [213](#)
- using copy pools [145](#)
- using data set separation [317](#), [320](#)
- using NaviQuest [329](#), [341](#)
- using Naviquest test cases [328–331](#), [368](#)
- using sharing control data sets [239](#)
- using the RLS\_MAX\_POOL\_SIZE parameter [243](#)
- using VOL=REF vs. VOL=SER [165](#)

#### record, data class attributes [118](#)

#### recording technology [130](#)

#### Recorg [129](#)

#### recover processing [170](#)

#### recovering

- ACDS [213](#)

#### recovering (*continued*)

- COMMDS [213](#)
- control data sets [213](#)
- SCDS [213](#)
- SMS address space [213](#)

#### recovery

- location [138](#)
- SMS
  - address space [215](#)
  - complex [215](#)
  - information [213](#), [215](#)
- VSAM RLS [260](#)

#### reducing false lock contention [246](#)

#### REFRESH command

- ISMF [302](#)

#### regression test [324](#)

#### regression testing, comparing [331](#)

#### RELEASE command

- ISMF [302](#)

#### remote access [312](#)

#### rename data set, name in use [227](#)

#### reports, creating [385](#)

#### requirements

- changing SMS configurations [183](#)
- defining shareoptions for data class [132](#)
- defining storage classes for VSAM RLS in a JES3 environment [251](#)
- defining the base configuration [20](#)
- defining the CF lock structure [244](#)
- determining hardware requirements for system-managed lock structure duplexing [236](#)
- determining hardware requirements for VSAM RLS [235](#)
- displaying storage group status using ISMF [190](#), [191](#)
- enabling extend processing [26](#)
- enabling VSAM RLS processing [252](#)
- manually activating the first SMS configuration [179](#)
- restarting the SMSVSAM server [261](#)
- specifying an extend storage group [37](#)
- using ACS routines [150](#)
- using NaviQuest [327](#), [328](#), [330](#), [336](#)
- using read-write variables in ACS routines [269](#)
- using the COPY BYPASSACS command [160](#)
- using the RESTORE BYPASSACS command [161](#)

#### reserve storage pool

- defining [55](#)

#### reserve storage pools

- assigning volumes [55](#)

#### RESHOW command

- ISMF [302](#)

#### resource class, FIELD [232](#)

#### resource owner [217](#)

#### restarting OAM [184](#)

#### RESTORE command

- ISMF [302](#)

#### restrictions

- allocating control data sets [11](#), [12](#), [14](#)
- assigning DASD storage groups to data sets [44](#)
- defining data class attributes [126](#)
- defining object class transition attributes [79](#)
- defining the base configuration [20](#)
- information recovery [213](#)
- managing VSAM data sets allocated using IDCAMS [298](#)
- modifying caching characteristics [310](#)
- modifying destaging characteristics [311](#)

## restrictions (*continued*)

- processing data class attributes in JCL [136](#)
- reinitializing a storage subsystem [311](#)
- running SMS in a Parallel Sysplex environment [6](#)
- SMS volume selection for data set allocation [101](#)
- specifying attributes for BWO and recovery [133](#)
- testing ACS routines [157](#)
- using ACS routines [150](#)
- using aggregate groups [137](#)
- using control data sets [213](#)
- using Navquest test cases [368](#)
- using OAM read-only variables [282](#)
- using read-only variables [283](#)
- using SMF data with NaviQuest [325](#)
- using system-managed lock structure duplexing [260](#)
- using the CF for VSAM record-level sharing [99](#)
- using the FILTLIST VALID\_UNITS INCLUDE statement [298](#)
- using the RLS\_MaxCfFeatureLevel parameter [257](#)
- using the SETSMS operator command [182](#), [183](#), [203](#)
- using VMA data with NaviQuest [326](#)

results, expected [324](#)

retain days extra backup versions [77](#)

retain days only backup version [77](#)

retained locks [245](#)

retention limit [70](#)

retention period

- limit [70](#)
- management class comparison [72](#)
- read-only variable [279](#)

Retpds (data class field) [121](#)

retrieving ISMF variables [387](#)

RETURN command

- ISMF [302](#)

REUSE [15](#), [16](#)

RIGHT command

- ISMF [302](#)

RLS Above the 2-GB Bar parameter [132](#)

RLS CF cache value parameter [132](#)

RLS\_MAXCfFeatureLevel

- IGDSMSxx member [248](#)

RLsAboveTheBarMaxPoolSize

- IGDSMSxx member [248](#)

RLSINIT parameter [263](#)

rolled-off GDS [76](#)

running test cases [331](#)

## S

### SAVE command

- ISMF [302](#)

### saving

- expected results [333](#)
- ISMF variables [387](#)

### SCDS

- ACS routine validation [155](#)
- allocation [15](#)
- backup [12](#)
- base configuration
  - installation defaults [19](#)
- base configuration definition [19](#), [23](#), [24](#)
- changing SMS configurations [183](#)
- copying object storage group [52](#)
- coupling facility definition [23](#), [24](#), [250](#)

## SCDS (*continued*)

- creating [12](#)
- defining status [20](#)
- definition [11](#)
- error [15](#)
- JCL allocation [15](#)
- modifying [12](#)
- multivolume [15](#)
- recovery [213](#)
- relationship with ACDS [12](#)
- saving [14](#)
- size calculation [13](#)
- SMS configuration [11](#)
- system group names [19](#)
- system names [19](#)
- updating [12](#), [208](#)

SCDS Status field [20](#)

scenario, testing [334](#)

SDR

- value [91](#), [92](#)

SECLABEL read-only ACS variable

- specifying in the ACS routine [176](#)

SECLABEL value

- ACS test case [178](#)
- RACF profile [175](#)

SECLABL read-only ACS variable

- overview [173](#)

SECMODEL parameter [173](#), [176](#)

secondary list [101](#)

secondary space reduction (SSR) [126](#)

security label

- read-only variable [279](#)

security labels in ACS routines

- creating a security label in the RACF profile [175](#)
- extracting the security label from the RACF profile [176](#)
- installation tasks [174](#)
- overview [173](#)
- planning tasks [174](#)
- specifying the &SECLABL read-only variable [176](#)
- writing ACS test cases using SECLABEL [178](#)

SELECT statement [294](#)

selection data set

- aggregate group field [208](#)
- description [138](#)

sequential data striping

- batch processing [91](#)
- sustained data rate [91](#)

serialization error exit [79](#)

SET operator command (T SMS command) [17](#), [179](#)

SET statement [292](#)

SETCACHE function [309](#)

SETCACHE line operator [309](#)

SETOAM statement [28](#)

SETOSMC statement [67](#)

SETROPTS command [176](#)

SETSMS operator command [15](#), [17](#), [180](#), [182](#)

setup, testing environment [327](#)

SETXCF START command [246](#), [255](#), [256](#)

Shareoptions parameter [131](#)

SHAREOPTIONS parameter

- specified for an SHCDS [239](#)

sharing

- CF cache structures [242](#)

sharing control data set (SHCDS) [239](#)



- SHCDS
  - recovering [261](#)
  - share options [239](#)
- SHCDS (sharing control data set) [239](#)
- shortcut keys [395](#)
- size
  - CF cache structure [242](#)
  - CF lock structure [244](#)
  - SHCDS [240](#)
- SMF records
  - CF-related [258](#)
  - interval time [182](#)
  - test case creation [330](#)
- SMS
  - 32-name support [14](#)
  - activation [11](#)
  - active
    - configuration information [185](#)
    - data set information [313](#)
    - volume information [313](#)
  - address space
    - canceling [215](#)
  - aggregate group [1](#)
  - base configuration [1, 3](#)
  - cache statistics [189](#)
  - capacity planning information [313](#)
  - classes, determining [172](#)
  - complex [3, 15, 22](#)
  - complex recovery [215](#)
  - configuration
    - activating [179](#)
    - automatic activation [180](#)
    - changing parameters [182](#)
    - content [3](#)
    - manual activation [179](#)
    - minimal [4](#)
  - construct naming convention [5](#)
  - converting volumes to SMS [212](#)
  - copying classes, storage groups, aggregate groups, copy pools [208](#)
  - definition [1](#)
  - deleting classes [211](#)
  - device status [192](#)
  - IGDSMSxx [17](#)
  - information recovery [213](#)
  - initialization [11](#)
  - initialization parameters [17](#)
  - IPL of SMS complex systems [179](#)
  - management class, default [21](#)
  - management policy [3, 5](#)
  - OAM DB2 ID [48](#)
  - OAM deletion protection [48](#)
  - OAM retention protection [48](#)
  - operator commands
    - DISPLAY SMS command [185](#)
    - VARY SMS command [185](#)
  - Parallel Sysplex [6, 22](#)
  - Parallel Sysplex environment [1](#)
  - physical storage [25](#)
  - pre-ACS interface [162](#)
  - preliminary steps [11](#)
  - preparing [5](#)
  - protecting [217](#)
  - RACF
- SMS (*continued*)
  - RACF (*continued*)
    - function [218](#)
    - STGADMIN profiles [226](#)
  - recovering
    - ACDS [213](#)
    - address space [213](#)
    - COMMDS [213](#)
    - control data sets [213](#)
    - SCDS [213](#)
  - resource owner [218](#)
  - SMS class [1](#)
  - storage group
    - deleting [209](#)
    - determining [172](#)
    - LISTSYS line operator [189](#)
    - moving volumes [203](#)
    - redefining volumes [203](#)
  - storage group status [47, 50](#)
  - storage management policy [1](#)
  - system
    - group [6, 22](#)
    - name [6, 22](#)
    - system-managed storage [1](#)
    - TRACE information [186](#)
    - volume status [192](#)
- SMS class
  - naming [4](#)
- SMS complex
  - definition [6](#)
  - system group [6](#)
- SMS configuration
  - ACS routine [3](#)
  - activating [231](#)
  - base configuration [11](#)
  - 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 parallel access volume (PAV) option
  - defining new storage classes [111](#)
  - displaying information about [112](#)
  - error messages [113](#)
  - installation tasks [110](#)
  - modifying storage classes [111](#)
- SMS volume selection
  - space constraint relief [109](#)
- SMSVSAM
  - terminate address space [264](#)
- SMSVSAM address space
  - recovering [261](#)
- SnapShot [83, 84, 93, 143](#)
- solid state drives
  - defining MSR for [88](#)
- SORT command
  - ISMF [302](#)
- sort criteria
  - aggregate group [202](#)
  - data class [201](#)
  - management class [202](#)

- sort criteria (*continued*)
  - optical
    - drive [202](#)
    - library [202](#)
  - storage
    - class [202](#)
    - group [202](#)
- source control data set (SCDS)
  - SMS configuration [11](#)
- source volume [93](#)
- space
  - attributes [118](#)
  - constraints [126](#), [128](#)
  - use information [313](#)
- space constraint relief [127](#)
- sphere, disconnect on CLOSE [99](#)
- statistics
  - active configuration [185](#)
  - cache [189](#)
  - coupling facility monitoring [259](#)
  - device [192](#)
  - OAM [185](#)
  - storage group
    - LISTSYS line operator [189](#)
  - volume [192](#)
- STGADMIN profiles
  - command and keyword related [226](#)
- storage administrator
  - authorizing access [233](#)
  - control data set application selection [19](#), [23](#), [24](#), [250](#)
  - ISMF primary option menu [2](#)
  - primary option menu
    - accessing [18](#)
- storage class
  - accessibility attributes [93](#)
  - ACS routine sample [298](#)
  - altering [1](#), [205](#)
  - application selection panel [84](#), [98](#), [99](#), [251](#)
  - assigning [100](#)
  - attributes
    - sequential data striping [83](#)
    - VSAM RLS [83](#)
  - availability attributes [85](#), [92](#)
  - bias [89](#)
  - continuous availability [30](#)
  - copying SMS [208](#)
  - coupling facility [84](#)
  - DASD data sets
    - system-managed [83](#)
  - data set placement [102](#)
  - defining [1](#), [84](#)
  - defining additional [101](#)
  - definition [251](#)
  - deleting [211](#)
  - description [4](#), [83](#)
  - device failure [92](#)
  - displaying [1](#)
  - displaying list [202](#)
  - dual copy [93](#)
  - guaranteed space [95](#)
  - initial access response seconds [90](#)
  - millisecond response [85](#), [88](#), [89](#)
  - objects [84](#)
  - performance [85](#)
- storage class (*continued*)
  - planning [84](#)
  - point-in-time copy attributes [93](#)
  - protection [232](#)
  - RACF [232](#)
  - sort criteria [202](#)
  - tape data sets [84](#)
  - tape volumes [84](#)
  - understanding [83](#)
  - view criteria [202](#)
  - volume allocation [97](#)
  - VSAM RLS [98](#), [251](#)
- Storage Class Alter/Define panel
  - defining storage classes with the PAV option enabled [111](#)
  - error message [113](#)
  - modifying storage classes to enable the PAV option [111](#)
- Storage Class Display panel
  - displaying PAV capability [112](#)
- storage control unit [309](#)
- storage group
  - ACS routine [97](#), [172](#)
  - adding volumes [42](#)
  - additional groups [52](#)
  - altering [203](#)
  - application selection panel [32](#)
  - assigning [51](#)
  - attributes [32](#), [34](#)
  - auto migrate [35](#)
  - automatic interval migration [35](#)
  - backup object attributes [48](#)
  - changing [203](#)
  - convert status [26](#)
  - copy pool backup
    - attributes [45](#)
    - defining [45](#)
  - copying SMS [208](#)
  - DASD volume
    - SMS complex [43](#)
    - status [26](#)
  - defining [25](#), [32](#), [35](#)
  - defining volume status [42](#)
  - deleting [209](#)
  - description [4](#)
  - displaying list [202](#)
  - dummy
    - attributes [33](#), [44](#)
    - defining [44](#)
    - volume serial numbers [44](#)
  - dump classes [38](#)
  - eligible volumes [91](#)
  - extend [37](#)
  - LIST commands [200](#)
  - location [27](#)
  - migration threshold [38](#)
  - multiple [28](#)
  - name [33](#)
  - naming [4](#)
  - OAM collection names [29](#)
  - object
    - attributes [46](#)
    - backup [34](#)
    - directory [31](#)
    - storage hierarchy [27](#)



- storage group (*continued*)
  - object backup [25](#), [34](#), [48](#)
  - on-demand migration [35](#)
  - optical volumes [50](#)
  - overflow [37](#)
  - planning [30](#)
  - planning for OAM [30](#), [31](#)
  - pool [25](#), [33–35](#), [37](#)
  - primary space [91](#)
  - qualifier [27](#)
  - sort criteria [202](#)
  - status
    - LISTSYS line operator [189](#)
    - volume [42](#)
  - status field [42](#)
  - Status field [52](#)
  - storage panels [34](#)
  - system access [40](#)
  - tape
    - defining [51](#)
    - library definition [29](#)
    - SMS support [29](#)
  - type [25](#), [34](#)
  - types [27](#)
  - view criteria [202](#)
  - VIO
    - defining [34](#)
  - volume
    - characteristics [25](#)
    - moving [203](#)
    - redefining [203](#)
- storage group ACS routine
  - specifying &SECLABL read-only variable [176](#)
  - testing [178](#)
- Storage Management Subsystem (SMS) [1](#)
- striped data sets [26](#), [30](#)
- striping volume selection [105](#)
- subtype
  - prefix [324](#)
  - test set [324](#)
  - testing next [333](#)
- summary of changes xxv
- sustained data rate (SDR) [91](#)
- SYS1.PARMLIB
  - COUPLExx member [22](#)
  - creating [5](#)
  - IEASYSy member [17](#)
  - IEFSSNxx member [17](#)
  - IGDSMSxx member [17](#), [248](#)
  - initializing [17](#)
  - modifying [5](#), [11](#), [17](#)
  - SMS [5](#)
  - SMS complex [5](#)
- system group
  - common configuration [6](#)
  - definition [6](#)
  - name [6](#)
- system name
  - definition [6](#)
- system-managed
  - data sets [4](#)
  - objects [4](#)

## T

- T SMS command [17](#), [179](#)
- tape library
  - SMS support [29](#)
- tape management system support [168](#)
- tape mount management [36](#), [38](#), [326](#)
- TAPEBLKSZLIM keyword [130](#)
- target volume [93](#)
- tasks
  - (noun, gerund phrase)
    - step [19](#)
  - using data set separation
    - specifying the data set separation profile [21](#)
- terminate, SMSVSAM [264](#)
- terminology, new [323](#)
- tertiary list [101](#)
- test bed library [324](#)
- test case, definition [324](#)
- test cases
  - generating [329](#)
  - running [331](#)
- test data restrictions [326](#)
- test results, validating [332](#)
- test set, subtype [324](#)
- testbed library attributes [328](#)
- testing
  - ACS routine [157](#)
  - ACS routines [327](#)
  - environment setup [327](#)
  - initial [324](#)
  - next subtype [333](#)
  - phase set [324](#)
  - running ACS routine cases [158](#)
  - scenario [334](#)
- TOP command
  - ISMF [302](#)
- track images [93](#)
- trademarks [400](#)
- transition copy technique [79](#)
- translation
  - ACS routine
    - description [151](#)
    - running with mixed levels of DFSMS [156](#)
- tree diagram [326](#)
- TSO/E (time sharing option), authorizing access [233](#)

## U

- UCB [193](#)
- UNBOX [193](#)
- unit affinity [166](#)
- unlisted device [195](#)
- UP command
  - ISMF [302](#)
- updating
  - FILTLISTs [330](#)
- user interface
  - ISPF [395](#)
  - TSO/E [395](#)
- user profile, RACF [175](#)
- user-defined variable [281](#)

## V

- VALIDATE command
  - ISMF [303](#)
- validating
  - test results [332](#)
- validation
  - ACS routine
    - conditions [154](#)
    - entire SCDS [155](#)
  - browsing ACS routine results [155](#)
  - individual ACS routine [153](#)
- variables, read-only
  - ACS routine [270](#)
  - definition [270](#)
  - initializing [281](#)
  - list [270](#)
  - OAM [282](#)
  - restrictions [283](#)
- variables, read-write
  - definition [269](#)
  - list [269](#)
- VARY SMS command [259](#)
- versioning devices [93](#)
- VIEW command
  - ISMF [303](#)
- VIEW CRITERIA (data class field) [201](#)
- VIO
  - ALLOCxx considerations [35](#)
  - Maxsize [35](#)
  - storage group allocation [33](#)
  - unit [34](#)
- virtual concurrent copy [84](#), [93](#)
- Vital Record Specification (VRS) [69](#)
- VMA (volume mount analyzer)
  - Navquest test case [370](#)
- volume
  - data class [121](#)
  - DISPLAY SMS command [192](#)
  - dummy storage group [44](#)
  - selection
    - enhanced [102](#)
    - rules [106](#)
  - serial number comparison [287](#)
  - status [192](#)
  - storage characteristics [25](#)
  - VSAM attributes [121](#)
- VOLUME [193](#)
- volume count
  - dynamic [124](#)
  - how SMS determines [127](#)
- volume full at write error [47](#), [49](#)
- volume full threshold [47](#), [49](#)
- volume serial label [199](#)
- volumes
  - assigning to a reserve storage pool [55](#)
- VSAM
  - attributes (data class) [126](#)
  - data class attributes [126](#)
  - data set [91](#)
  - linear data set
    - SMS.SCDS1.SCDS [15](#), [16](#)
  - Record-Level Sharing (RLS) [23](#)
  - storage groups [25](#)

- VSAM record-level sharing (RLS) [6](#)
- VSAM RLS
  - activating [252](#)
  - adjusting lock structure size [247](#)
  - administering [235](#)
  - authorization [249](#)
  - authorizing access [249](#)
  - base configuration [23](#)
  - CF cache and lock structures [242](#)
  - coupling facility (CF) [98](#)
  - displaying CF information [253](#)
  - environment [236](#)
  - fallback processing [262](#)
  - false lock contention [246](#)
  - hardware requirements
    - coupling facility (CF) [235](#)
  - lock structure [244](#)
  - monitoring the coupling facility (CF) [253](#)
  - planning for availability [238](#)
  - preparing [235](#)
  - recovery [260](#)
  - requirements [252](#)
  - retained locks [245](#)
  - SHCDS
    - defining [239](#)
  - storage class definition [251](#)
  - SYS1.PARMLIB IGDSMSxx member
    - modifying [248](#)
  - system connectivity requirements [237](#)
- VT (Virtual Tape Server)
  - initial access response seconds [90](#)

## W

- WRITE statement [295](#)

## X

- XCF [6](#)
- XES coupling definition process [244](#)
- XRC [83](#)

## Z

- zHyperLinks
  - defining use of, by storage class [99](#)





Product Number: 5655-ZOS

SC23-6860-70

