

DFSMS/MVS Version 1 Release 5



DFSMSrmm Guide and Reference

DFSMS/MVS Version 1 Release 5



DFSMSrmm Guide and Reference

Note!

Before using this information and the product it supports, be sure to read the "Notices" on page xvii.

Eighth Edition (March 1999)

This edition applies to Version 1 Release 5 of DFSMS/MVS (5695-DF1), Release 7 of OS/390 (5645-001), and any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

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

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
RCF Processing Department
G26/050
5600 Cottle Road
SAN JOSE, CA 95193-0001
U.S.A.

Or you can send your comments electronically to starpubs@vnet.ibm.com.

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

© **Copyright International Business Machines Corporation 1992, 1999. All rights reserved.**

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

Contents

Figures	xi
Notices	xvii
Programming Interface Information	xvii
Trademarks	xviii
About This Book	xix
Required Product Knowledge	xix
How to Use This Book	xix
DFSMSrmm Publications	xx
Referenced Publications	xx
How to Tell If a Book Is Current	xx
Notational Conventions	xxi
How to Read Syntax Diagrams	xxi
How to Abbreviate Commands and Operands	xxiv
How to Use Continuation Characters	xxiv
Delimiters	xxiv
How to Use the Labeled Boxes	xxiv
References to Product Names Used in DFSMS/MVS Publications	xxiv
Summary of Changes	xxvii
Publication Updates	xxvii
Eighth Edition, March 1999	xxvii
Seventh Edition, June 1998	xxvii
Sixth Edition, June 1997	xxviii
Fifth Edition, March 1997	xxviii
Fourth Edition, September 1996	xxviii
Third Edition, Small Programming Enhancement, August 1995	xxix
Chapter 1. Introducing DFSMSrmm	1
What Libraries and Locations Can DFSMSrmm Manage?	1
What Is in a Removable Media Library?	1
What Is in a System-Managed Tape Library?	2
What Is in a Non-System-Managed Tape Library?	3
What Is in a Storage Location?	3
How Does DFSMSrmm Manage These Libraries and Locations?	4
Setting Up Your Installation Options	4
Defining Retention and Movement Policies	5
Running DFSMSrmm Utilities	8
What Resources Does DFSMSrmm Manage?	8
Shelf Locations	9
Volumes	10
Data Sets	13
Year 2000 Support	14
Software Products	14
Owner Information	14
How Does DFSMSrmm Help You Create Reports?	14
Using the DFSMSrmm Application Programming Interface	14
Using DFSMSrmm ISPF Dialog and RMM TSO Subcommands	15
EDGAUD and EDGRPTD Report Utilities	15
Using the DFSMSrmm EDGRRPTE Exec.	15
Using the DFSORT ICETOOL Utility	16
How Does DFSMSrmm Authorization and Security Work?	16

What Tape Usage Does DFSMSrmm Support?	16
How Does DFSMSrmm Validate Tape Mounts?	17
Tape Volume Rejection by DFSMSrmm	18
Rejects Caused by Installation Controls	18
Rejects Caused by Validation Failure	18
Rejects Caused by DFSMSrmm Rules	19
Who Can Use DFSMSrmm and How?	20
General User	20
Tape Librarian	20
Storage Administrator	20
System Programmer	20
Operator	21
Using DFSMSrmm	21
 Chapter 2. Quick Reference for Basic DFSMSrmm Tasks.	23
Using the DFSMSrmm Dialog	23
Changing Dialog and Processing Options.	23
Using Fast Path Commands	24
Requesting Help	24
Exiting the Dialog	24
Getting Information about Your Resources	24
Sorting Lists	25
Printing Display Panels and Lists	26
Requesting Lists and Resource Details in Batch	27
Requesting a Scratch Volume	28
Releasing a Volume Manually	29
Changing Owner Information	31
Changing Volume Information	31
Changing Release Date for a Volume	34
Setting Release Actions for a Volume	34
Changing Data Set Information	35
Requesting Notification of a Volume's Release	37
Requesting Return of a Volume after Release	38
Requesting Special Volume and Data Set Actions.	38
Retaining Data Sets and Volumes	38
Grouping Your Volumes	38
 Chapter 3. Getting Started with DFSMSrmm.	39
Starting the Dialog	39
Starting DFSMSrmm from an ISMF Selection Menu	39
Starting DFSMSrmm from an ISPF Primary Option Panel	40
Using the RMMISPF EXEC	40
Setting Dialog Options.	41
Changing Processing Options	41
Changing List Sorting Options	44
Using Dialog Panels and Fast Path Commands	46
Navigating Through DFSMSrmm	46
Using Menus	47
Issuing Fast Path Commands	49
Using Help Panels	50
Exiting the Dialog	50
 Chapter 4. Defining Your Resources.	53
Defining Shelf Locations	53
Adding New Rack and Bin Numbers.	54
Deleting Rack and Bin Numbers	55

Defining Volumes	56
Adding a New, Non-Scratch Volume.	57
Adding Scratch Volumes	59
Redefining a Volume Already Defined to DFSMSrmm	60
Changing Volume Information	60
Deleting Volume Information	62
Defining Owners	63
Adding New Owners	63
Changing Owner Information	64
Deleting Owner Information	65
Defining Software Products and Product Versions.	66
Adding a New Software Product	66
Adding Software Product Volumes	68
Changing Software Product Information	69
Deleting Software Product Information	71
Defining Data Sets	72
Adding New Data Set Information	73
Changing Data Set Information	73
Deleting Data Set Information	76
Chapter 5. Defining Vital Record Specifications	79
Types of Vital Record Specifications.	79
Data Set Vital Record Specifications	79
Name Vital Record Specifications.	80
Volume Vital Record Specifications	80
Chaining Vital Record Specifications	80
Identifying Data Sets for Retention	81
Fully Qualified Data Set Names	82
Data Set Name Masks.	82
Management Class and Management Value Names	83
Special Data Set Name Masks.	83
Name Filtering with Data Set Name Masks	83
How DFSMSrmm Applies Retention and Movement Policies	83
Generation Data Group (GDG) Base Names	88
Pseudo-GDG Data Set Names.	88
Specifying GDG and Non-GDG Data Set Names	89
Identifying Volumes for Retention.	90
Retaining Volumes by Specific Volume Serial Number	90
Retaining Volumes by Generic Volume Serial Number	90
Adding Vital Record Specifications	90
Adding Data Set Vital Record Specifications.	92
Adding Volume Vital Record Specifications	93
Adding Name Vital Record Specifications	94
Changing Vital Record Specifications	94
Deleting a Vital Record Specification	96
Chapter 6. Retaining and Moving Your Volumes	99
Types of Retention	99
CYCLES.	99
BYDAYSCYCLE	100
DAYS	100
EXTRADAYS	100
LASTREFERENCEDAYS.	100
WHILECATALOG.	100
UNTILEXPIRED	101
By Data Set Open Status.	101

By Job Name	102
By Specific Volume Serial Number	102
By Generic Volume Serial Number	102
By Specific Date	102
Defining Retention Policies for Data Sets and Volumes.	103
Defining Policies for Management Class and Management Value	104
Combining Retention Policies in a Vital Record Specification.	105
Merging Retention Policies	105
Defining System-Wide Vital Record Specifications	106
DFSMSrmm Retention Dates	106
Retention Date Format.	107
Special Cycles Date Format.	107
Special Catalog Date Format	107
Special CATRETPD Date Format.	108
How DFSMSrmm Calculates the Retention Date	108
Data Set Retention Date	108
Volume Retention Date	109
Retaining Data Sets in Closed-Cycle GDGs	112
Managing Closed-Cycle GDGs.	112
Converting from Closed-Cycle GDGs to Using Scratch Pools	113
Defining Movement Policies for Volumes	113
Moving between System-Managed Libraries.	114
Managing the Contents of System-Managed Libraries	114
Moving to Storage Locations	116
Using Manual Move Control.	117
Manually Requesting Volume Moves	118
Requesting a Single Volume Move	118
Moving Volumes to a System-Managed Tape Library	120
Ejecting Volumes from a System-Managed Tape Library	120
Moving Volumes Between Pools	122
Moving Volumes to Loan Locations	122
Returning Volumes from Loan Locations	123
Moving Volumes from Storage Locations	123
Confirming Volume Movements to DFSMSrmm.	124
Confirming Volume Moves into an Automated Tape Library	124
Confirming Volume Moves for Other Locations	124
Chapter 7. Defining Retention and Movement Policies	129
Before You Start	129
A Summary of the Tasks To Perform	129
Where to Find Information	129
Which Type of Vital Record Processing Should You Use?	130
What is Your Level of Experience with DFSMSrmm?	130
Selecting VRSEL(OLD) Processing	130
Selecting VRSEL(NEW) Processing	131
Migrating to Using VRSEL(NEW).	132
Chaining Retention and Movement Policies	133
What You Can Chain When You Specify VRSEL(OLD)	133
What You Can Chain When You Specify VRSEL(NEW).	134
Specifying Policies	134
Defining the Minimum Required Policy	136
Writing Your Own Vital Record Specifications	136
Example 1 — Retaining Catalogued Data Sets.	136
Example 2 — Retaining Uncatalogued Data Sets	137
Example 3 — Retaining Cycles of Data Sets	137
Example 4 — Moving Data Sets to Different Locations	137

Example 5 — Retaining Data Sets Created within a Time Period	138
Example 6 — Holding Data Sets for Extra Days	138
Example 7 — Retaining Generation Data Group Data Sets	139
Example 8 — Retaining Recently Used Data Sets	139
Example 9 — Using Management Value to Retain Data Sets	139
Example 10 — Returning Volumes to Scratch Status	140
Example 11 — Retaining Data Sets Using Expiration Date	140
Example 12 — Combining Vital Record Specifications	141
 Chapter 8. Storage Location Management for System-managed Tape	
Virtual Tape Servers	143
DFSMSrmm Export Processing	143
DFSMSrmm Volume Not In Library Support	144
DFSMSrmm Import Processing	144
Completing the IMPORT Process.	145
 Chapter 9. Requesting and Releasing Volumes	147
Requesting Scratch Volumes Manually.	147
Releasing Volumes	148
Setting Release Actions for a Volume	149
Changing a Volume's Release Date	154
Confirming Manual Release Actions to DFSMSrmm	155
Releasing Volumes Manually	158
 Chapter 10. Requesting Information about Your Resources.	163
Using DFSMSrmm Reporting Utilities and Sample Reports	163
Displaying Details about a Single Resource	163
Shelf Location Details	166
Volume Details	166
Owner Details	168
Software Product Details	168
Data Set Details	169
Vital Record Specification Details.	170
Displaying DFSMSrmm System Options	171
Using Resource Lists	172
How to Request a List	172
How to Sort a List	176
How to Use Line Operators	182
Requesting Specific Types of Lists	185
Scratch Volume Pull Lists	185
Drop/Ship Lists	186
Lists of Pools Defined by Your Installation	188
Lists of Volume Movements and Actions	189
CLISTs of Executable Subcommands	191
Printing Resource Information	191
 Chapter 11. DFSMSrmm Operator Procedures	193
Restarting the DFSMSrmm Subsystem.	193
Using the START Command	193
Implementing Subsystem Parameter Changes	194
Stopping the DFSMSrmm Subsystem	195
Controlling the Problem Determination Aid (PDA) Facility	196
Responding to DFSMSrmm Operator Messages	196
Initialization Messages.	197
Tape Mount Messages.	200
Tape Validation Messages	202

Tape Processing Messages	205
System Error Messages	206
DFSMSrmm Installation Exit Messages	207
Initializing and Erasing Tape Volumes Manually	208
Using the LABEL Procedure to Request EDGINERS Processing	209
Replying to LABEL Procedure Messages	212
INIT and ERASE Request Syntax	215
Processing Sticky Labels	217
Labels.	218
 Chapter 12. Using RMM TSO Subcommands	 221
Issuing the RMM TSO Command and Subcommands	221
Requesting Help for RMM TSO Subcommands	222
Submitting a Batch Job	222
ADDBIN: Adding a Bin Number in a Storage Location	224
ADDDATASET: Adding Data Set Information.	227
ADDDOWNER: Adding Owner Information	233
ADDPRODUCT: Adding Software Product Information	236
ADDRACK: Adding a Shelf Location.	238
ADDVOLUME: Adding Volume Information	241
ADDVRS: Adding a Vital Record Specification	255
CHANGEDATASET: Changing Data Set Information	270
CHANGEOWNER: Changing Owner Information	277
CHANGEPRODUCT: Changing Software Product Information	280
CHANGEVOLUME: Changing Volume Information	282
DELETEBIN: Deleting Bin Number Information	308
DELETEDATASET: Deleting Data Set Information.	311
DELETEDOWNER: Deleting Owner Information	313
DELETEPRODUCT: Deleting Software Product Information	315
DELETERACK: Deleting Shelf Location Information	317
DELETEVOLUME: Deleting Volume Information	320
DELETEVRS: Deleting Vital Record Specifications	324
GETVOLUME: Requesting and Assigning Scratch Volumes	326
LISTBIN: Displaying Information about a Shelf Location	331
LISTCONTROL: Displaying Parmlib Options and Control Information.	333
LISTDATASET: Displaying Information about a Data Set	342
LISTOWNER: Displaying Information about an Owner	345
LISTPRODUCT: Displaying Information about a Software Product.	347
LISTRACK: Displaying Information about a Shelf Location	349
LISTVOLUME: Displaying Information about a Volume	351
LISTVRS: Displaying Information about a Vital Record Specification	355
SEARCHBIN: Creating a List of Bin Numbers	358
SEARCHDATASET: Creating a List of Data Sets	362
SEARCHPRODUCT: Creating a List of Software Products	369
SEARCHRACK: Creating a List of Shelf Locations	372
SEARCHVOLUME: Creating a List of Volumes.	376
SEARCHVRS: Creating a List of Vital Record Specifications.	392
 Chapter 13. DFSMSrmm Return and Reason Codes	 401
RMM TSO Subcommand Return Codes	401
RMM TSO Subcommand Reason Codes	402
 Chapter 14. Creating REXX Execs.	 411
TSO Subcommand Variables by Subcommand.	411
TSO Subcommand Variables by Name.	415
Sample REXX Execs	426

EDGXMP1 VOLCHAIN EXEC	426
EDGXMP2 DSNLIST EXEC.	429
Appendix. DFSMSrmm ISPF Dialog Fast Path Commands	433
Abbreviations	435
Glossary	439
Index	449
Readers' Comments — We'd Like to Hear from You.	465

Figures

1. Example of the DFSMSrmm DELETEVOLUME Syntax Diagram	xxiii
2. Defining a Vital Record Specification Chain	7
3. Example of a List of Volumes Owned by a Single User	15
4. DFSMSrmm Librarian Menu	22
5. DFSMSrmm User Menu	23
6. DFSMSrmm Volume List Sort Options Panel	26
7. DFSMSrmm Request a Volume Panel	28
8. DFSMSrmm Release Volumes Panel	29
9. DFSMSrmm Confirm Volume Release Panel	30
10. Example of Volume List	30
11. DFSMSrmm Change Owner Details Panel	31
12. DFSMSrmm Volume Search Panel	32
13. DFSMSrmm Change Volume Details Panel	33
14. DFSMSrmm Data Set Search Panel	35
15. Data Set List Example	36
16. DFSMSrmm Change Data Set Details Panel	37
17. DFSMSrmm Primary Option Menu	39
18. RMMISPF Exec Syntax Diagram	40
19. DFSMSrmm Dialog Options Menu	41
20. DFSMSrmm Dialog User Options Panel	42
21. DFSMSrmm Dialog Sort Options Menu	45
22. DFSMSrmm Data Set List Sort Options Panel	45
23. DFSMSrmm Command Menu	48
24. DFSMSrmm Volume Menu	49
25. DFSMSrmm Volume Display Panel	50
26. DFSMSrmm Exit Menu	51
27. DFSMSrmm Add Racks and Bins Panel	54
28. DFSMSrmm Delete Racks and Bins Panel	56
29. DFSMSrmm Add Volume Panel	58
30. DFSMSrmm Add Scratch Volumes Panel	59
31. DFSMSrmm Change Volume Panel	61
32. DFSMSrmm Change Volume Details Panel	62
33. DFSMSrmm Add Owner Details Panel	64
34. DFSMSrmm Change Owner Details Panel	65
35. DFSMSrmm Delete Owner Panel	66
36. DFSMSrmm Add Product Panel	67
37. DFSMSrmm Add Product Details Panel	67
38. DFSMSrmm Add Product Volume Panel	68
39. DFSMSrmm Change Product Panel	69
40. DFSMSrmm Change Product Details Panel	70
41. DFSMSrmm Delete Product Panel	71
42. DFSMSrmm Confirm Delete Product Panel	72
43. DFSMSrmm Add Data Set Details Panel	73
44. DFSMSrmm Change Data Set Panel	74
45. DFSMSrmm Change Data Set Details Panel	75
46. DFSMSrmm Delete Data Set Panel	76
47. DFSMSrmm Confirm Delete Data Set Panel	77
48. Chaining Vital Record Specifications	81
49. Specifying a Name Vital Record Specification for a Chain	81
50. Specifying a Vital Record Specification Chain	81
51. Valid Job Name Masks	86
52. Defining a Retention Policy for a GDG Base Name	88
53. Specifying Data Set Name Masks for Pseudo-GDGs	88

54. Specifying NOGDG and CYCLES in a Vital Record Specification	89
55. Specifying NOGDG and DAYS in a Vital Record Specification	90
56. DFSMSrmm Add Vital Record Specification Panel	91
57. DFSMSrmm Add Data Set VRS Panel	92
58. DFSMSrmm Add Volume VRS Panel	93
59. DFSMSrmm Add Name VRS Panel	94
60. DFSMSrmm Change Vital Record Specification Panel	95
61. DFSMSrmm Change Data Set VRS Panel	96
62. DFSMSrmm Delete Vital Record Specification Panel	97
63. DFSMSrmm Delete Name VRS Panel	98
64. Retaining a Data Set Using Management Class	104
65. Retaining a Data Set Using Management Value	104
66. Using UNTILEXPIRED and WHILECATALOG on a Single Vital Record Specification	105
67. Merging UNTILEXPIRED with the WHILECATALOG Retention Policy.	106
68. Managing Closed-Cycle GDGs	113
69. Managing Volumes Using Scratch Pools	113
70. Moving Volumes out of a System-Managed Library	115
71. Moving a Volume in and out of the System-managed Library.	116
72. Moving Data out of the System-managed Library at Fixed Intervals	116
73. Keeping Volumes On-site	117
74. Sending a Volume to Another System	117
75. Returning a Volume from Another System.	118
76. DFSMSrmm Change Volume Details Panel	119
77. Ejecting a Volume from a System-managed Library	121
78. Moving a Volume to a Loan Location	123
79. Supplying Loan Location Information for Volumes	123
80. Returning a Volume from a Loan Location.	123
81. DFSMSrmm Volume Action Status Panel	125
82. DFSMSrmm Volume Action Summary List.	126
83. Confirming Volume Moves for Volumes Ready to Scratch	127
84. Identifying Locations and Destinations for Volumes	127
85. Specifying a Vital Record Specification With Default Operands	134
86. Specifying a Vital Record Specification Using the Dialog Part 1	135
87. Specifying a Vital Record Specification Using the Dialog Part 2	135
88. Creating a Volume Export List	143
89. Confirming Volume Moves for Exported Volumes	144
90. Creating a Volume Import List	145
91. DFSMSrmm Request a Volume Panel	147
92. Release Actions on the DFSMSrmm Add Volume Panel	150
93. EDGPC700 - DFSMSrmm Volume Action Status Panel	156
94. DFSMSrmm Volume Action Summary List.	157
95. Confirming Replace Actions for Volumes	158
96. DFSMSrmm Release Volumes Panel	159
97. DFSMSrmm Confirm Volume Release Panel.	160
98. DFSMSrmm Confirm Volume Eject Panel	160
99. DFSMSrmm List of Volumes.	161
100. DFSMSrmm Volume Display Panel	165
101. DFSMSrmm Rack or Bin Details Panel	166
102. DFSMSrmm Volume Details Panel	166
103. DFSMSrmm Volume Details Panel (Cont.)	167
104. DFSMSrmm Owner Details Panel.	168
105. DFSMSrmm Product Details Panel	169
106. DFSMSrmm Data Set Details Panel	169
107. DFSMSrmm Display Data Set VRS Panel.	170
108. DFSMSrmm Display Volume VRS Panel	170

109. DFSMSrmm Display Name VRS Panel	171
110. DFSMSrmm System Options Display	172
111. DFSMSrmm Rack and Bin Search Panel	174
112. DFSMSrmm Dialog Sort Options Menu	176
113. DFSMSrmm Data Set List Sort Options Panel	177
114. SORT Command Syntax Diagram for Rack or Bin Number Lists	180
115. SORT Command Syntax Diagram for Volume Lists	181
116. SORT Command Syntax Diagram for Lists of Volume Movements and Actions	181
117. SORT Command Syntax Diagram for Software Product.	181
118. SORT Command Syntax Diagram for Data Set Lists	182
119. SORT Command Syntax Diagram for Rack or Bin Number Lists	182
120. Producing a List of Scratch Volumes.	185
121. DFSMSrmm Rack and Bin Search Panel	186
122. DFSMSrmm Volume Action Status Panel	187
123. DFSMSrmm Volume Action Summary List.	187
124. DFSMSrmm Control Information Menu Panel	189
125. Example of a List of Pools	189
126. DFSMSrmm Volume Action Summary List.	190
127. START Command Syntax Diagram	193
128. Command Syntax for Restarting DFSMSrmm and Implementing New Parmlib Options	194
129. Command Syntax for Refreshing DFSMSrmm Installation Exits	194
130. Command Syntax for Shutting Down DFSMSrmm	195
131. Command Syntax for Quiescing DFSMSrmm	195
132. Command Syntax for Removing DFSMSrmm Using the EDGRESET Utility	196
133. LABEL Command Syntax Diagram	209
134. INIT and ERASE Request Syntax Diagram	215
135. Default Label Format for a Tape Cartridge.	218
136. Default Label Format for a Round Tape.	218
137. ADDBIN Syntax Diagram	225
138. ADDDATASET Syntax Diagram	228
139. ADDOWNER Syntax Diagram	234
140. ADDPRODUCT Syntax Diagram	236
141. ADDRACK Syntax Diagram	239
142. ADDVOLUME Syntax Diagram	242
143. ADDVOLUME Non-SCRATCH Parameters	243
144. ADDVOLUME Non-SCRATCH Parameters (Cont.)	244
145. ADDVRS DATASET Syntax Diagram	257
146. ADDVRS Location NAME Syntax Diagram	258
147. ADDVRS Retention NAME Syntax Diagram	259
148. ADDVRS VOLUME Syntax Diagram	260
149. CHANGEDATASET Syntax Diagram Part 1	271
150. CHANGEDATASET Syntax Diagram Part 2	272
151. CHANGEOWNER Syntax Diagram	277
152. CHANGEPRODUCT Syntax Diagram	280
153. CHANGEVOLUME Syntax Diagram for Global Movement and Confirmrelease Parameters	286
154. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters	287
155. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters (Cont.)	288
156. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters (Cont.)	289
157. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters (Cont.)	290
158. DELETEBIN Syntax Diagram	309

159. DELETEDATASET Syntax Diagram	311
160. DELETEOWNER Syntax Diagram.	313
161. DELETEPRODUCT Syntax Diagram.	315
162. DELETERACK Syntax Diagram	318
163. DELETEVOLUME Syntax Diagram	321
164. DELETEVRS Syntax Diagram	324
165. GETVOLUME Syntax Diagram	327
166. Sample LISTBIN Output	331
167. LISTBIN Syntax Diagram	331
168. Sample LISTCONTROL Output.	334
169. LISTCONTROL Syntax Diagram	335
170. Sample LISTCONTROL Output.	338
171. Sample LISTCONTROL Output.	339
172. LISTCONTROL LOCDEF Output - No LOCDEFs Defined	340
173. LISTCONTROL LOCDEF Output - LOCDEFs Defined	340
174. Sample LISTDATASET Output	343
175. LISTDATASET Syntax Diagram	343
176. Sample LISTOWNER Output	345
177. LISTOWNER Syntax Diagram	345
178. Sample LISTPRODUCT Output	347
179. LISTPRODUCT Syntax Diagram	347
180. Sample LISTRACK Output	349
181. LISTRACK Syntax Diagram	349
182. Sample LISTVOLUME Output	352
183. LISTVOLUME Syntax Diagram	352
184. Sample LISTVOLUME Output	353
185. Sample LISTVRS Output	355
186. LISTVRS Syntax Diagram	356
187. Listing Vital Record Specification Information Examples.	356
188. Listing Vital Record Specification Information	356
189. Sample SEARCHBIN Output	358
190. SEARCHBIN Syntax Diagram	359
191. Sample SEARCHDATASET Output	362
192. SEARCHDATASET Syntax Diagram	363
193. Sample SEARCHDATASET Output	367
194. Sample SEARCHDATASET Output	367
195. Sample SEARCHDATASET Output	367
196. Sample CLIST Data Set	368
197. Sample SEARCHPRODUCT Output	369
198. SEARCHPRODUCT Syntax Diagram	370
199. Sample SEARCHRACK Output.	372
200. SEARCHRACK Syntax Diagram	373
201. Sample SEARCHVOLUME Output Listing Owned Volumes	377
202. SEARCHVOLUME Syntax Diagram Part 1	378
203. SEARCHVOLUME Syntax Diagram Part 2	379
204. SEARCHVOLUME Syntax Diagram Part 3	380
205. Sample SEARCHVOLUME Output Listing Volumes to Eject	388
206. SEARCHVOLUME Output Using CLIST Operand	389
207. Sample SEARCHVOLUME Output Using RETDATE Operand	389
208. Sample SEARCHVOLUME Output Building a List of Logical Volumes	389
209. Sample SEARCHVOLUME Output Identifying Logical Volumes to DFSMSrmm.	390
210. Moving Volumes from One Library to Another Library	390
211. Confirming Volume Moves from One Library to Another Library	390
212. Sample SEARCHVRS Output	393
213. SEARCHVRS Syntax Diagram	394

214. Sample SEARCHVRS Output	399
215. Sample SEARCHVRS Output	399
216. Testing RMM TSO LISTVOLUME Subcommand Example 1	401
217. Testing RMM TSO LISTVOLUME Subcommand Example 2	401
218. VOLCHAIN EXEC Sample REXX Exec.	427
219. DSNLIST EXEC Sample REXX Exec	430

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

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 of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

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
Information Enabling Requests
Dept. DWZ
5600 Cottle Road
San Jose, CA 95193

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this publication to non-IBM Web sites are provided for convenience only, and do not in any manner serve as an endorsement of these Web sites.

Programming Interface Information

This publication is intended to help you use DFSMSrmm.

This publication also documents General-use Programming Interface and Associated Guidance Information provided by DFSMSrmm.

General-use programming interfaces allow the customer to write programs that obtain the services of DFSMSrmm.

General-use Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Trademarks

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

AnyNet	MVS
BookManager	MVS/DFP
CICS	MVS/ESA
CUA	NetView
DFSMSdfp	OPC
DFSMSdss	OpenEdition
DFSMShsm	OS/2
DFSMSrmm	OS/390
DFSMS/MVS	RACF
DFSORT	RMF
ESCON	Sysplex Timer
FFST/MVS	System/390
GDDM	SystemView
Hardware Configuration Definition	SOMobjects
Hiperspace	VisualLift
IBM	VTAM
IMS	
Language Environment	

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About This Book

This book is intended for tape librarians, storage administrators, operators, and system programmers responsible for handling and managing removable media. It is also intended for general users who use removable media.

This book helps you:

- Perform tasks, such as defining resources to DFSMSrmm, releasing volumes, requesting scratch volumes, defining policies for retaining and moving data sets and volumes, creating lists, and displaying information recorded by DFSMSrmm
- Use the DFSMSrmm ISPF Dialog to manage and use your removable media
- Use the RMM TSO command set to manage and use your removable media

Required Product Knowledge

You should be familiar with these before using DFSMSrmm:

ISPF, for using the DFSMSrmm ISPF dialog

TSO, for using the TSO command, RMM, and related subcommands

How to Use This Book

For information on using DFSMSrmm, see the following:

“Chapter 1. Introducing DFSMSrmm” on page 1

“Chapter 2. Quick Reference for Basic DFSMSrmm Tasks” on page 23 provides step by step instructions for using DFSMSrmm for the general user

“Chapter 3. Getting Started with DFSMSrmm” on page 39

“Chapter 4. Defining Your Resources” on page 53

“Chapter 5. Defining Vital Record Specifications” on page 79 describes how to define vital record specifications

“Chapter 6. Retaining and Moving Your Volumes” on page 99 describes the retention and movement policies you can define with DFSMSrmm

“Chapter 7. Defining Retention and Movement Policies” on page 129

“Chapter 8. Storage Location Management for System-managed Tape Virtual Tape Servers” on page 143¹

“Chapter 9. Requesting and Releasing Volumes” on page 147

“Chapter 10. Requesting Information about Your Resources” on page 163

“Chapter 11. DFSMSrmm Operator Procedures” on page 193 provides information for the operator to respond to system messages and to initialize and erase tapes

“Chapter 12. Using RMM TSO Subcommands” on page 221

“Chapter 13. DFSMSrmm Return and Reason Codes” on page 401

“Chapter 14. Creating REXX Execs” on page 411

1. Import/Export support is available with APAR OW36342 or OW36343.

The appendix contains:

“Appendix. DFSMSrmm ISPF Dialog Fast Path Commands” on page 433

DFSMSrmm Publications

The following publications have additional information about DFSMSrmm:

Publication Title	Order Number
<i>DFSMS/MVS DFSMSrmm Command Reference Summary</i>	SX26-6016
<i>DFSMS/MVS DFSMSrmm Diagnosis Guide</i>	SY27-9615
<i>DFSMS/MVS DFSMSrmm Implementation and Customization Guide</i>	SC26-4932
<i>DFSMS/MVS DFSMSrmm Application Programming Interface</i>	SC26-7272

Referenced Publications

This book refers to the following publications:

Publication Title	Order Number
<i>DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries</i>	SC26-3051
<i>DFSMS/MVS Using ISMF</i>	SC26-4911
<i>OS/390 ISPF User's Guide</i>	SC28-1239
<i>OS/390 MVS Planning: Operations</i>	GC28-1760
<i>OS/390 MVS System Messages, Vol 2 (ASB-EWX)</i>	GC28-1785
<i>OS/390 TSO/E User's Guide</i>	SC28-1968

How to Tell If a Book Is Current

IBM regularly updates its books with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a book are identical, but subsequent updates may be available in softcopy before they are available in hardcopy. Here's how to determine the level of a book:

- Check the book's order number suffix (often referred to as the dash level). A book with a higher dash level is more current than one with a lower dash level. For example, in the publication order number SC26-4930-02, the dash level 02 means that the book is more current than previous levels, such as 01 or 00. Suffix numbers are updated as a product moves from release to release, as well as for hardcopy updates within a given release.
- Check to see if you are using the latest softcopy version. To do this, compare the last two characters of the book's file name (also called the book name). The higher the number, the more recent the book. For example, DGT1U302 is more recent than DGT1U301.
- Compare the dates of the hardcopy and softcopy versions of the books. Even if the hardcopy and softcopy versions of the book have the same dash level, the softcopy may be more current. This will not be apparent from looking at the edition notice. The edition notice number and date remain that of the last hardcopy version. When you are looking at the softcopy product bookshelf, check the date shown to the right of the book title. This will be the date that the softcopy version was created.

Also, an asterisk (*) is added next to the new and changed book titles in the CD-ROM booklet and the README files.

Vertical lines to the left of the text indicate changes or additions to the text and illustrations. For a book that has been updated in softcopy only, the vertical lines indicate changes made since the last printed version.

Notational Conventions

This section explains the notational conventions used in this book.

How to Read Syntax Diagrams

Throughout this library, diagrams are used to illustrate the programming syntax. Keyword parameters are parameters that follow the positional parameters. Unless otherwise stated, keyword parameters can be coded in any order. The following list tells you how to interpret the syntax diagrams:

- Read the diagrams from left-to-right, top-to-bottom, following the main path line. Each diagram begins on the left with double arrowheads and ends on the right with two arrowheads facing each other.

▶▶ | Syntax Diagram | ◀◀

- If a diagram is longer than one line, each line to be continued ends with a single arrowhead and the next line begins with a single arrowhead.

▶▶ | First Line | ▶

▶▶ | Second Line | ▶

▶▶ | Last Line | ▶

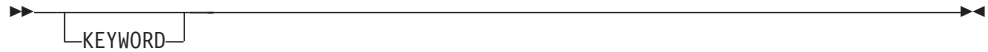
- Required keywords and values appear on the main path line. You must code required keywords and values.

▶▶ — REQUIRED_KEYWORD — ▶▶

If several mutually exclusive required keywords or values exist, they are stacked vertically in alphanumeric order.

▶▶ —
REQUIRED_KEYWORD_OR_VALUE_1
REQUIRED_KEYWORD_OR_VALUE_2 — ▶▶

- Optional keywords and values appear below the main path line. You can choose not to code optional keywords and values.



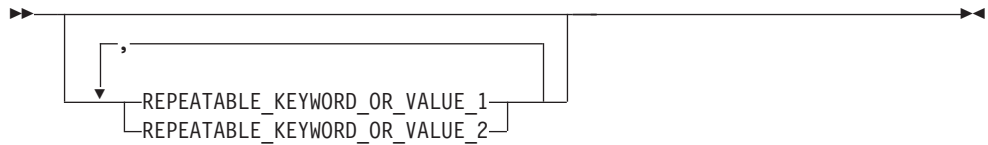
If several mutually exclusive optional keywords or values exist, they are stacked vertically in alphanumeric order below the main path line.



- An arrow returning to the left above a keyword or value on the main path line means that the keyword or value can be repeated. The comma means that each keyword or value must be separated from the next by a comma.



- An arrow returning to the left above a group of keywords or values means more than one can be selected, or a single one can be repeated.



- A word in all uppercase is a keyword or value you must spell exactly as shown. In this example, you must code **KEYWORD**.



If a keyword or value can be abbreviated, the abbreviation is discussed in the text associated with the syntax diagram.

- If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code **KEYWORD=(001,0.001)**.



- If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code **KEYWORD=(001 FIXED)**.



- Default keywords and values appear above the main path line. If you omit the keyword or value entirely, the default is used.



- A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

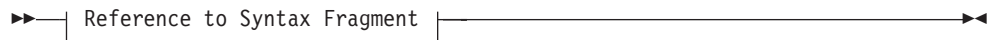


- References to syntax notes appear as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

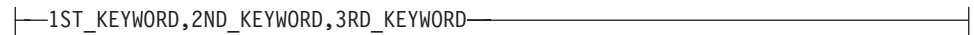


Notes:

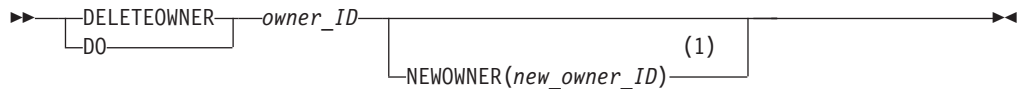
1. An example of a syntax note.
- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.



Syntax Fragment:



The following figure shows an example of a syntax diagram.



Notes:

1. Must be specified if the owner owns one or more volumes.

Figure 1. Example of the DFSMSrmm DELETEVOLUME Syntax Diagram

The possible valid versions of the RMM DELETEOWNER command are:

```
RMM DELETEOWNER owner
RMM DO          owner
RMM DELETEOWNER owner NEWOWNER(new_owner)
RMM DO          owner NEWOWNER(new_owner)
```

How to Abbreviate Commands and Operands

The TSO abbreviation convention applies for all DFSMSrmm commands and operands. The TSO abbreviation convention requires you to specify as much of the command name or operand as is necessary to distinguish it from the other command names or operands.

Some DFSMSrmm keyword operands allow unique abbreviations. All unique abbreviations are shown in the command syntax diagrams.

How to Use Continuation Characters

The symbol - is used as the continuation character in this book. You can use either - or +.

- Do not ignore leading blanks on the continuation statement
- + Ignore leading blanks on the continuation statement

Delimiters

When you type a command, you must separate the command name from the first operand by one or more blanks. You must separate operands by one or more blanks or a comma. Do not use a semicolon as a delimiter because any character you enter after a semicolon is ignored.

How to Use the Labeled Boxes

In the sections of this book that describe using the DFSMSrmm ISPF dialog, we provide labeled boxes identifying the related TSO subcommands you can use to perform the same task. Here is an example of a labeled box:

Related TSO Subcommand

Use the ADDOWNER subcommand to define a new owner to DFSMSrmm. See “ADDOWNER: Adding Owner Information” on page 233 for more information on ADDOWNER.

References to Product Names Used in DFSMS/MVS Publications

DFSMS/MVS publications support DFSMS/MVS, 5695-DF1, as well as the DFSMSdfp base element and the DFSMSshsm, DFSMSdss, and DFSMSrmm features of OS/390, 5647-A01. DFSMS/MVS publications also describe how DFSMS/MVS interacts with other IBM products to perform the essential data, storage, program and device management functions of the operating system.

DFSMS/MVS publications typically refer to another IBM product using a generic name for the product. When a particular release level of a product is relevant, the reference includes the complete name of that product. This section explains the naming conventions used in the DFSMS/MVS library for the following products:

MVS can refer to:

- MVS/ESA SP Version 5, 5695-047 or 5695-048

- The MVS base control program (BCP) of OS/390, 5647-A01

All MVS book titles used in DFSMS/MVS publications refer to the OS/390 editions. Users of MVS/ESA SP Version 5 should use the corresponding MVS/ESA book. Refer to *OS/390 Information Roadmap* for titles and order numbers for all the elements and features of OS/390.

For more information about OS/390 elements and features, including their relationship to MVS/ESA SP and related products, please refer to *OS/390 Planning for Installation*.

RACF can refer to:

- Resource Access Control Facility (RACF), Version 2, 5695-039
- The RACF element of the OS/390 Security Server, an optional feature of OS/390

All RACF book titles refer to the Security Server editions. Users of RACF Version 2 should use the corresponding book for their level of the product. Refer to *OS/390 Security Server (RACF) Introduction* for more information about the Security Server.

CICS can refer to:

- CICS/MVS, 5665-403
- CICS/ESA, 5685-083
- The CICS element of the CICS Transaction Server for OS/390, 5665-147

All CICS book titles refer to the CICS Transaction Server for OS/390 editions. Users of CICS/MVS and CICS/ESA should use the corresponding books for those products. Please see *CICS Transaction Server for OS/390: Planning for Installation* for more information.

Summary of Changes

This section no longer includes product highlights. For similar information, please see these books:

- *DFSMS/MVS Planning for Installation*, SC26-4919. This book describes each functional enhancement and its requirements for implementation.
- *DFSMS/MVS General Information*, GC26-4900. "What's New in Version 1 Release 5" provides an overview of each enhancement.

Publication Updates

The following sections describe specific publication updates to this book.

Eighth Edition, March 1999

This publication is a minor revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 5. This book was last published June 1998. Vertical lines to the left of the text and illustrations indicate technical changes or additions since the June 1998 version. This version also includes maintenance and editorial changes. The changes made in this revision include information about:

- DFSMSrmm disposition processing
- DFSMSrmm ISO/ANSI V4 support
- DFSMSrmm catalog status tracking
- RMM TSO subcommand enhancements
- DFSMSrmm virtual tape server import/export processing support²

Seventh Edition, June 1998

This publication is a major revision in support of the functional changes provided by vital record specification processing enhancements. Vertical lines to the left of the text and illustrations indicate technical changes or additions since the June 1997 version. This version also includes maintenance and editorial changes.

- Managing a data set with more than one retention policy.
- Retaining all the copies of a data set created on the same day as a single cycle.
- Defining a period beyond the date when the DFSMSrmm policy started to retain the data set.
- Avoiding unnecessary movement by specifying that a volume stays in its current location.
- Managing a data set or volume using a combination of retention and movement policies.

2. Import/Export support is available with APAR OW36342 or OW36343.

- Releasing volume and returning them to scratch status in one run of release processing by using the RELEASE operand for volumes that have no release actions pending.
- Releasing volumes based on the retention policy rather than the volume expiration date specified in the JCL or set by the installation.

This publication also supports:

- Enhancements to the RMM TSO commands that allow the user to change data set and volume information recorded by DFSMSrmm during O/C/EOV processing. The enhancements also allow the user to change the ddname and the job stepname of the job that created the data set.

Sixth Edition, June 1997

This publication is a minor revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 4. This book was last published March 1997, in softcopy only.

This revision also includes maintenance and editorial changes.

- Removed information about changing panel color and emphasis options
- Updated information on how DFSMSrmm calculates retention date
- Updated operator message examples including operator responses to Problem Determination Aid messages
- Updated RMM ADDVRS subcommand information
- Updated RMM CHANGEVOLUME subcommand information
- Changed the MAXRETPD parmlib default value

Fifth Edition, March 1997

This publication is a minor revision in support of the following items:

- Improvements have been made to DFSMSrmm inventory management vital records processing.
- Vital record information is maintained for data sets even when they are no longer managed by a vital record specification.

Fourth Edition, September 1996

This publication is a major revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 3. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change. For a book that has been updated in softcopy only, the vertical lines indicate changes made since the last printed version.

This revision also includes maintenance and editorial changes.

The following summarizes the changes to that information.

- DFSMSrmm ISPF dialog panel examples have been changed to reflect consistency with ISPF Version 4 panels.
- DFSMSrmm TSO subcommands have been updated.
- Global movement and release confirmation information has been added.
- New support for storing JOB and STEP accounting information has been added.

- Installation control over the use of DFSMSrmm subcommands information has been added.
- New support for managing data sets and volumes information has been added.
- Back up of the DFSMSrmm control data set using DFSMSdss concurrent copy to minimize impact to the tape environment
- Support for 3590 Tape Subsystem

Third Edition, Small Programming Enhancement, August 1995

This publication is a minor revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 2.

This revision also includes maintenance and editorial changes.

The following summarizes the changes to that information.

- Implementation of installation defined storage locations
- Use of jobname for selection of policy
- Retention of volumes in a location until expired
- Removal of three storage locations limit
- Storage locations without BINS
- Subdivision of storage location BINS by MEDIANAME
- Ability to separate movement reports by volume status
- Manual override of vital record specification movement
- Ability to cancel volume moves
- Support for no label output requests to scratch tapes, and extended support for changing volume labels at output time.

Chapter 1. Introducing DFSMSrmm

In your enterprise, you store and manage your removable media in several types of media libraries. For example, in addition to your traditional tape library, a room with tapes, shelves, and drives, you might have several automated and manual tape libraries. You probably also have both on-site libraries and off-site storage locations, also known as vaults or stores.

With the DFSMSrmm functional component of DFSMS/MVS, you can manage your removable media as one enterprise-wide library across systems. DFSMSrmm manages your installation's tape volumes and the data sets on those volumes. DFSMSrmm also manages the shelves where volumes reside in all locations except in automated tape library dataservers.

DFSMSrmm manages all tape media, such as cartridge system tapes and 3420 reels, as well as other removable media you define to it. For example, DFSMSrmm can record the shelf location for optical disks and track their vital record status; it does not manage the objects on optical disks.

The following sections discuss some basic tape management concepts, introduce the terminology used throughout the DFSMSrmm publications and in the DFSMSrmm Interactive System Productivity Facility (ISPF) dialog, and discuss tape management tasks you can perform with DFSMSrmm.

What Libraries and Locations Can DFSMSrmm Manage?

You decide where to store your removable media based on how often the media is accessed and for what purpose it is retained. For example, you might keep volumes that are frequently accessed in an automated tape library datsarver, and you probably use at least one storage location to retain volumes for disaster recovery and audit purposes. You might also have locations where volumes are sent for further processing such as other data centers within your company or your customers and vendors. This section defines the libraries and storage locations DFSMSrmm can manage and the loan locations to which volumes might be sent. This section also helps you select where to store your removable media.

DFSMSrmm can manage:

- Removable media library, which incorporates all other libraries, such as:
 - System-managed tape libraries; for example, the automated IBM 3494 Tape Library Dataserver and IBM 3495 Tape Library Dataserver models, and the manual IBM 3495 M10 Tape Library Dataserver.
 - Non-system-managed tape libraries, or traditional tape libraries
- Storage locations, both on-site and off-site

What Is in a Removable Media Library?

A *removable media library* contains all the tape and optical volumes that are available for immediate use, including the shelves where they reside. A removable media library usually includes other libraries: *system-managed libraries* such as *automated* or *manual tape library dataservers*; and *non-system-managed libraries*, containing the volumes, shelves, and drives not in an automated or a manual tape library datsarver.

In the removable media library, you store your volumes in shelves, where each volume occupies a single *shelf location*. This shelf location is referred to as a *rack number* in the RMM TSO subcommands and DFSMSrmm ISPF dialog. A rack number matches the volume's external label. DFSMSrmm uses the external volume serial number to assign a rack number when adding a volume, unless you specify otherwise. The format of the volume serial you define to DFSMSrmm must be one to six alphanumeric characters. The rack number must be six alphanumeric or national characters.

What Is in a System-Managed Tape Library?

A *system-managed tape library* is a collection of tape volumes and tape devices defined in the tape configuration database. The tape configuration database is an integrated catalog facility user catalog marked as a volume catalog (VOLCAT) containing tape volumes and tape library records. A system-managed tape library can be either automated or manual.

You can have several automated or manual tape library dataservers; you use an installation-defined library name to define each automated tape library datserver or manual tape library datserver to the system and DFSMSrmm treats each system-managed tape library as a separate location or destination. See *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for additional information.

Automated Tape Library Dataservers

An *automated tape library datserver* is a device consisting of robotic components, cartridge storage areas (or shelves), tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. The IBM automated tape libraries are the automated IBM 3494 and IBM 3495 Tape Library Dataservers. The IBM 3494 Tape Library Datserver supports 3490E and 3590 tape subsystems. The automated models of the IBM 3495 Tape Library Datserver support 3490, 3490E, and 3590 tape subsystems. The manual model of the IBM 3495 Tape Library Datserver support 3490 and 3490E tape subsystems. The IBM 3494 Tape Library Datserver complements the IBM 3495 Tape Library Datserver by offering a smaller-capacity and lower-price tape automation solution. ³The IBM 3494 Tape Library Datserver can also include the Virtual Tape Server subsystem.

DFSMSrmm can automatically replenish the scratch volumes in an automated tape library datserver when the supply of volumes becomes low by reclaiming volumes that are eligible for returning to scratch. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information about replenishing volumes in an automated tape library datserver.

Manual Tape Library Dataservers

A *manual tape library datserver* is a set of tape drives and the set of system-managed volumes the operator can mount on those drives. The IBM manual tape library is the manual IBM 3495 Tape Library Datserver, which supports 3490 and 3490E Magnetic Tape Subsystems.

3. Import/Export support is available with APAR OW36342 or OW36343.

DFSMSrmm has a cartridge entry installation exit that can be used to help partition volumes in a single system-managed tape library across multiple systems. This can include both VM and MVS sharing, and also partitioning between MVS systems and SMS complexes.

- Support for partitioning with VM is based on volume naming conventions, and at the individual volume level for volumes defined to DFSMSrmm on MVS, using the DFSMSrmm parmlib options to identify volumes to be partitioned.
- With a single tape configuration data base across multiple MVS systems and complexes, you also require a single DFSMSrmm control data set. All volumes can be used on any or all systems with no partitioning possible.

See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for implementation details for these scenarios.

What Is in a Non-System-Managed Tape Library?

A *non-system-managed tape library* is all the volumes, shelves, and drives not in an automated tape library dataserver or manual tape library dataserver. You might know this library as the traditional “tape library”. DFSMSrmm provides complete tape management functions for the volumes and shelves in this traditional tape library. Volumes in a non-system-managed library are defined by DFSMSrmm as being *shelf resident*.

All tape media and drives supported by MVS/ESA are supported in this environment. Using DFSMSrmm, you can fully manage all types of tapes in a non-system-managed tape library, including 3420 reels, 3480, and 3590 cartridge system tapes.

You can also use DFSMSrmm to manage volumes in an IBM Tape Library Data server which is managed with Basic Tape Library Support (BTLS) or any other automated tape library which has special software.

What Is in a Storage Location?

Storage locations are not part of the removable media library because the volumes in storage locations are not generally available for immediate use. A storage location is comprised of shelf locations that you define to DFSMSrmm. A shelf location in a storage location is identified by a *bin number*. Storage locations are typically used to store removable media that are kept for disaster recovery or vital records. DFSMSrmm manages two types of storage locations: installation defined storage locations and DFSMSrmm built-in storage locations.

You can define an unlimited number of installation defined storage locations, using any eight-character name for each storage location. You can also define the type or shape of the media in the location and the bin numbers that DFSMSrmm assigns to the shelf locations in the storage location. You can request DFSMSrmm shelf-management when you want DFSMSrmm to assign a specific shelf location to a volume in the location.

You can also use the DFSMSrmm built-in storage locations, LOCAL, DISTANT, and REMOTE. Although the names of these locations imply their purpose, they do not mandate their actual location. All volumes can be in the same or separate physical location. For example, an installation could have the LOCAL storage location onsite, as a vault in the computer room, the DISTANT storage location could be a vault in an adjacent building, and the REMOTE storage location is a secure facility across

town or in another state. DFSMSrmm provides shelf-management for storage locations which means storage locations can be managed at the shelf location level.

Although DFSMSrmm automatically shelf-manages built-in storage locations, you must first define the bins you want to use to DFSMSrmm. For bin numbers in built-in storage locations, the numbers are fixed in range, starting at bin number 000001. For installation defined storage locations, you can use any alphanumeric characters.

How Does DFSMSrmm Manage These Libraries and Locations?

DFSMSrmm records the complete inventory of the removable media library and storage locations in the DFSMSrmm control data set—a VSAM key-sequenced data set. In the control data set, DFSMSrmm records all changes made to the inventory, such as adding or deleting volumes, and also keeps track of all movement between libraries and storage locations.

DFSMSrmm manages the movement of volumes among all library types and storage locations. This lets you control where a volume, and hence a data set, resides and how long it is retained.

DFSMSrmm helps you manage the movement of your volumes and retention of your data over their full life, from initial use to the time they are retired from service. Among the functions DFSMSrmm performs for you are:

- Automatically initializing and erasing volumes.
- Recording information about volumes and data sets as they are used.
- Expiration processing.
- Identifying volumes with high error levels that require replacement.

To make full use of all of the DFSMSrmm functions, you specify installation setup options and define retention and movement policies. DFSMSrmm provides you with utilities to implement the policies you define. The following sections describe options, policies, and utilities.

Setting Up Your Installation Options

The DFSMSrmm parmlib member EDGRMMxx includes many options for setting up DFSMSrmm, such as:

- Defining system options, such as the date format for reports and messages, default retention periods, and whether to notify volume owners when their volumes are ready to be released.
- Preventing a range of tapes from being used on specific systems.
- Defining pools, such as the range of shelves to use for a pool, and whether a pool has RACF tape profile processing.
- Tailoring mount messages with either the volume's shelf location or the pool identifier.
- Defining security classes for data sets and volumes.
- Defining the DFSMSrmm running mode to determine when DFSMSrmm records volume usage and performs tape validation as described in “How Does DFSMSrmm Validate Tape Mounts?” on page 17.
- Defining how DFSMSrmm bypass label processing is performed.
- Defining storage locations to DFSMSrmm.

Your system programmer or storage administrator defines your DFSMSrmm installation options during implementation. For information on the options that can be set in EDGRMMxx, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

Defining Retention and Movement Policies

The retention and movement policies you define to DFSMSrmm are known as *vital record specifications*. You use them to specify how long and where you want to keep data sets or volumes. You also use them to define how volumes are to be moved among the libraries DFSMSrmm supports, and the storage locations defined for vital records and disaster recovery purposes.

Use vital record specifications to control retention requirements for production data in accordance with the service level agreement and known requirements.

Use the DFSMSrmm parmlib member EDGRMMxx to set options to control vital record processing and to set expiration and retention for your installation. Use the parmlib OPTION command VRCHANGE, VRSMIN, and VRSEL operands to control how DFSMSrmm processes your retention and movement policies. See “Which Type of Vital Record Processing Should You Use?” on page 130 for information on using the parmlib OPTION VRSEL operand to select vital record processing. When users are allowed to specify expiration and retention period to override vital record specifications, use the parmlib OPTION MAXRETPD operand to set a maximum retention period for your installation. See the *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information.

You can define policies, or vital record specifications, for data sets and volumes. You can also define special *name* vital record specifications to provide retention information for data sets and volumes that must be moved through multiple locations before they expire.

Defining Home Location and Target Destinations

DFSMSrmm records the starting location for a volume when the volume is initially defined to DFSMSrmm, or when volume information is changed. This starting location is known to DFSMSrmm as a *home* location. Home is where volumes start from and are returned to when the identified retention and movement actions have been completed. A non-system-managed library identified as SHELF can only be used as a home location and not as a target location in a vital record specification. SHELF can be used as a target location when a volume is moved by issuing a command.

You can give any system-managed library or storage location as a target destination for a volume move.

You can also use the DFSMSrmm-reserved location name CURRENT to avoid moving a volume from its current location.

Defining Retention Policies

Use data set names and data set name masks to define retention policies for data sets. Use job names and job name masks to define retention policies to further qualify the criteria for applying retention and movement policies. For data sets, you can request the following types of retention:

Retention by cycles by days

You can retain a minimum number of cycles or copies of a data set that are created on a single calendar day. DFSMSrmm manages these as a single cycle.

Retention by cycles

You can retain a minimum number of cycles or copies of a data set. This applies to generation data groups (GDGs) or like-named data sets identified by pseudo-GDG data set names. For non-GDG data sets, DFSMSrmm considers each occurrence of a data set to be a cycle.

Retention by number of elapsed days

You can retain copies of a data set that match a data set name mask produced during a given number of days.

Retention for a number of extra days

You can add extra days to the retention date for a data set if your installation specifies VRSEL(NEW).

Retention by days since last referenced

You can retain each copy of a data set produced for a set number of days since the data set was read or written.

Retention while data set is cataloged

You can retain any data set as long as it remains cataloged. Catalog status can also be used in combination with management by cycles, elapsed days, days since last referenced, and volume expiration date.

Retention to a specific date

You can set a deletion date for a vital record specification. When that date is reached, the vital record specification is deleted. All data sets and volumes that would match the vital record specification become eligible for release processing, or might match a less specific vital record specification that might specify different retention and movement information.

Retention by expiration date

You can retain the data set on a volume as long as the volume expiration date has not yet been reached. You can also use a combination of catalog status and volume expiration date to retain data sets.

Retention of open data sets

You can specify a separate policy to apply to all data sets that are currently open.

Retention of data sets closed by abend processing

You can specify a separate policy to apply to all data sets that were open at the time of an application or system abend.

You can also use a *vital record specification management value* instead of a data set name to define retention policies. A vital record specification management value assigns management and retention policies to tape data sets and is defined by your installation. You can define data set vital record specifications for vital record specification management values, to provide support for special JCL specified expiration dates, and to allow DFSMSrmm to manage those data sets with special dates. You can use the sample installation exit contained in SYS1.SAMPLIB, EDGUX100, to assign vital record specification management values.

You can use management class names when you are managing system-managed volumes. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information on assigning management class names and vital record specification management values using EDGUX100.

Defining Vital Record Specification Chains

You can create a vital record specification chain to define retention and movement policies. The first vital record specification in the chain is a data set or volume vital record specification. You can link vital record specifications to the first vital record specification to add additional retention and movement policies for a data set or volume. See “Chaining Retention and Movement Policies” on page 133 for information about chaining vital record specifications.

Figure 2 is an example of creating a vital record specification chain. The data set name vital record specification moves a data set from the installation media library to the REMOTE storage location. The name vital record specification, MOVE2, moves the volume from the REMOTE storage location to the DISTANT storage location.

```
RMM ADDVRS DSNAME(STEGO.SAURUS) COUNT(1855) DAYS LOCATION(REMOTE) -  
    STORENUMBER(30) NEXTVRS(MOVE2)  
RMM ADDVRS NAME(MOVE2) LOCATION(DISTANT) STORENUMBER(1825)
```

Figure 2. Defining a Vital Record Specification Chain

You can define retention and movement policies for specific volumes and volumes that match a generic volume serial number. If you use a specific volume serial number, DFSMSrmm retains the volume matching that volume serial number. If you use a generic volume serial number, DFSMSrmm retains a number of volumes matching the generic volume serial number based on the number you specify.

When multiple vital record specifications are defined for a volume, and each vital record specification contains a different destination, DFSMSrmm decides where to move the volume based on priority number. Table 1 shows the movement priority DFSMSrmm uses if you do not assign movement priority numbers. For example, if REMOTE and DISTANT are both defined as the destination for the volume, DFSMSrmm selects the REMOTE storage location as the destination. The selection is made based on the REMOTE priority number of 100 which is lower than the DISTANT priority number of 200.

Table 1. DFSMSrmm Movement Priority

Priority Number	Location Name or Location Type
100	REMOTE DFSMSrmm built-in storage location name
200	DISTANT DFSMSrmm built-in storage location name
300	LOCAL DFSMSrmm built-in storage location
2000	Installation defined storage locations
4800	AUTO automated tape libraries
4900	MANUAL manual tape libraries
5000	SHELF location name

Running DFSMSrmm Utilities

DFSMSrmm provides utilities to manage your inventory, create reports, maintain the DFSMSrmm control data set, and erase and initialize volumes.

Use the EDGHSKP utility to run inventory management activities.

- Vital record processing determines which data sets to retain and what volume moves are required, based on retention and movement policies defined to DFSMSrmm.

DFSMSrmm supports trial run and production run vital record processing. Trial run vital record processing does not change data set and volume information in the DFSMSrmm control data set. Using trial run vital record processing you have an opportunity to analyze the effect that your movement and retention policies will have. Based on your analysis, you can then determine if you need to change vital record specifications before performing production run vital record processing. This is helpful when you are defining your initial set of policies and when you need to make changes as you gain more experience with DFSMSrmm.

- Expiration processing identifies volumes ready to be released and returned to scratch.
- Storage location management processing assigns shelf locations to volumes being moved to storage locations.
- DFSMSrmm control data set functions back up the control data set and the journal, reset the journal data set when the control data set and journal are backed up, and create an extract data set.

Use the EDGAUD and EDGRPTD utilities and the EDGRRPTE exec to get information about your removable media library and storage locations. You can also get security trail information about volumes and data sets defined to DFSMSrmm, and audit trail information about volumes, shelf assignments, and user activity.

Use the EDGUTIL utility to create, update, and verify the control data set. Use the EDGBKUP utility to back up and recover the control data set and journal.

Use DFSMSrmm backup utilities instead of other backup utilities, such as access method services EXPORT, because DFSMSrmm provides the necessary serialization and forward recovery functions. DFSMSrmm backup utilities check whether the control data set is in use, tell the DFSMSrmm subsystem that backup or recovery is in process, and provide a way to forward recover using DFSMSrmm journal data sets.

Use the EDGINERS utility to erase and initialize tape volumes either automatically or manually. You can use EDGINERS instead of the DFSMSdfp utility IEHINITT. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for details on the differences between the utilities and for information on controlling the use of IEHINITT.

You can schedule these utilities to run at the time and frequency that are best for your installation. Use a scheduling system such as OPC/ESA for this purpose.

What Resources Does DFSMSrmm Manage?

The following sections describe how DFSMSrmm helps you manage shelf locations, volumes, data sets, information about volume owners, and software products in all libraries and storage locations.

Shelf Locations

DFSMSrmm automatically:

- Assigns the volume a rack number matching the volume serial number you specify, unless you specify a rack number or pool ID when you first define a volume to DFSMSrmm
- Assigns bin numbers for the shelf locations in shelf-managed storage locations
- Provides shelf location information, either a specific rack number or the rack number prefix of the first suitable pool, in the fetch/mount message to help the operator respond to mount requests

DFSMSrmm also helps you delete rack or bin numbers for obsolete, empty shelf locations. You can create lists containing information about the shelf locations in your removable media library and in storage locations.

You can have more control over where volumes reside and how they are managed by defining your removable media into pools. A pool is a group of shelf locations, defined by a common prefix, requiring DFSMSrmm pool management functions. For example, based on your pool definitions, DFSMSrmm can ensure RACF profiles exist for all volumes in a pool. You could also request that all expiration date protected volumes in a pool are processed automatically. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information about defining your pools in parmlib member EDGRMMxx. If you do not define any pools, DFSMSrmm considers all volumes part of one default pool.

In DFSMSrmm, there are two categories of pools:

Rack pool

A *rack pool* is shelf space that can be assigned to hold any volumes. Although you can add scratch volumes to these pools, you cannot normally use these volumes to satisfy non-specific mount requests.

Each rack pool can contain private volumes and scratch volumes:

- Private Volumes
 - Are not generally available for use because the data on them is not yet expired.
 - Can have a master status or user status.

A master status indicates that the volume contains data that can only be overwritten based on criteria set by the EDGRMMxx MASTEROVERWRITE operand.

A user status indicates that the volume contains data that can be overwritten without a data set name match.
- Scratch Volumes
 - Are available for use because they are either new, unused volumes, or they contain expired data.
 - Are used to satisfy scratch requests.

With scratch volumes in rack pools, you can provide a user or group-based pool of volumes, allowing selection using the RMM GETVOLUME subcommand or with the EDGUX100 installation exit. Scratch volumes in a rack pool cannot be used where DFSMSrmm system-based pooling is in use.

Scratch pool

A *scratch pool* is shelf space assigned to hold volumes for use with the

DFSMSrmm system based scratch pooling or exit-based pooling. The volumes assigned to this shelf space can be used to satisfy scratch requests as long as the volumes are in scratch status. Once the volume has been written to, it becomes a volume with MASTER status, until the volume is returned to scratch status.

Pools within system-managed libraries are based on the external volume serial number, which is also used as the rack number. Most volumes in a system-managed library belong to either a general scratch pool or they can be separated into multiple scratch pools, one for each media type. For example, volumes can be divided into one pool for cartridge system tape and one pool for enhanced capacity cartridge system tape. If volumes from other data centers are entered into a system-managed library for processing, a rack pool can be defined to allow you to take advantage of DFSMSrmm pool management functions.

Pools in non-system-managed libraries are logical sections of shelf space and also based on rack number. You can control shelf location assignment by specifying a pool ID when you define a volume to DFSMSrmm. DFSMSrmm automatically assigns the volume to the next available rack number in that pool. For volumes that you do not specify a shelf location or pool id, DFSMSrmm attempts to assign the rack number that matches the volume serial number.

Volumes

DFSMSrmm provides support for:

- Physical volumes, which are volumes that can be associated with a physical package which is independently mountable from other physical volumes.
- Logical volumes, which are volumes that reside in a Virtual Tape Server library either on DASD (in the tape volume cache as a virtual volume) or on a stacked volume (as a logical volume) and are referenced from the host as if they were physical volumes. Logical volumes can be removed from a VTS using export processing described in “Chapter 8. Storage Location Management for System-managed Tape Virtual Tape Servers” on page 143. DFSMSrmm does not currently manage stacked volumes while in a VTS but DFSMSrmm does record the stacked volume as the 'in container' value during the export process.⁴

DFSMSrmm automatically validates volumes, ensuring that only valid scratch volumes are mounted for non-specific mount requests, and that the right volume is mounted for a specific mount request. This eliminates unintentionally overwriting a valid master volume or a volume retained for disaster recovery or vital record management.

When a data set on a volume is opened and closed, DFSMSrmm automatically:

- Changes the volume status from scratch to master for non-specific mount requests at open time
- Sets an expiration date for the volume, and ensures that the maximum expiration date is not exceeded at open time
- Records information about data sets on the volume (data set name is recorded at open time; all other information is recorded at close time)
- Counts the number of times a volume is used since the volume was last in scratch status at open time

4. Import/Export support is available with APAR OW36342 or OW36343.

- Counts the number of temporary and permanent errors encountered at close time only
- Sets a security classification based on the data sets that reside on the volume at open time
- Prevents reading of data on a volume in scratch status when DFSMSrmm is running in protect mode at open time

DFSMSrmm automatically controls the movement of volumes within the removable media library and the built-in or installation defined storage locations, based on criteria you specify in vital record specifications.

When volumes no longer reside in the removable media library or storage locations, you can define loan locations where these volumes can be found. DFSMSrmm does not, however, manage the movement of volumes residing in loan locations.

DFSMSrmm automatically releases volumes that have reached their expiration date. Any non-scratch volume defined to DFSMSrmm has an expiration date indicating when the volume is to be considered for release. The volume expiration date can be:

- An expiration date or retention period, given in the JCL by the user when writing a data set to the volume.
- The date specified by the user when a scratch volume is requested using the RMM GETVOLUME subcommand or when information about the volume is manually added or changed.
- A default retention period for data sets set by your installation with the RETPD option in parmlib member EDGRMMxx in SYS1.PARMLIB.
- The expiration date that is set using the DFSMSrmm EDGUX100 installation exit. If the exit sets a zero date, the installation default retention period is used.

Your installation can also set a maximum retention period in parmlib member EDGRMMxx. An expiration date or retention period set for a volume cannot exceed this value.

You can retain a volume beyond its expiration date using a vital record specification to define a retention policy. You can also manually override an expiration date and release a volume before the expiration date is reached.

When a volume is eligible for release, DFSMSrmm can perform *release actions* for the volume based on information you provide. Examples of release actions include:

- Returning a volume to scratch status
- Initializing a volume
- Erasing a volume
- Notifying a volume's owner of the volume's release
- Returning a volume to its owner and deleting the volume record in the DFSMSrmm control data set

See “Releasing Volumes” on page 148 for information about setting release actions and how processing is performed.

DFSMSrmm tracks the number of I/O errors recorded for a volume. For permanent errors, DFSMSrmm automatically marks the volumes for replacement before returning them to scratch. This occurs during expiration processing.

You can get additional information to help you analyze volumes with temporary errors from these tools and services:

- DFSMSrmm records volume use and temporary I/O errors. You can obtain this information through the extract data provided by EDGHSKP.
- The Environmental Record Editing and Printing Program (EREP) uses LOGREC records to identify tapes with unacceptably high error conditions.
- SERVICE DIRECTOR identifies volumes with high error levels. See your IBM customer engineer for information about using this tool.
- System management facilities (SMF) records, produced by DFSMSdfp, track information on volume use, such as how much data is written to a volume, which you can use to analyze volumes with errors.

You can then use this information to identify volumes for replacement and tell DFSMSrmm to manage it. Use the RMM CHANGEVOLUME RELEASEACTION(REPLACE) subcommand or the dialog change volume function to tell DFSMSrmm to manage the replacement.

DFSMSrmm provides support for using RACF standard tape volume security protection using any combination of RACF TAPEVOL and TAPEDSN options.

DFSMSrmm allows you to use volumes with duplicate volume serial numbers as well as volumes undefined to DFSMSrmm. You can request that DFSMSrmm ignore a volume so it can be used, even if another volume with the same volume serial number is already defined in the DFSMSrmm control data set. If DFSMSrmm ignores a volume, it does not track volume use, and if the volume is not defined to DFSMSrmm, DFSMSrmm cannot provide any management functions for the volume.

DFSMSrmm Tape Label Support

DFSMSrmm supports the following tape label types:

- IBM standard labels (SL)
- ISO/ANSI labels (AL)
- Both IBM standard and user header or trailer labels (SUL)
- Both ISO/ANSI and user header or trailer labels (AUL)
- No labels (NL)

You can explicitly use AL, SL, and NL label types when you define volumes to DFSMSrmm. Although a tape is defined to DFSMSrmm as SL or AL, a user can still process it as SUL or AUL. A user could ask for a non-specific volume by specifying LABEL=(,SUL) and create an SL volume or SUL volume without affecting DFSMSrmm. DFSMSrmm records the value specified in the JCL.

DFSMSrmm provides support for ISO/ANSI Version 3 and ISO/ANSI Version 4 labels. You can provide the current ISO/ANSI label version of a volume that you define to DFSMSrmm. Then you can use the DFSMSrmm EDGINERS utility to initialize the tape with either ISO/ANSI Version 3 or ISO/ANSI Version 4 labels.

No label (NL) is supported for private volumes. Scratch tapes must be standard label tapes. DFSMSrmm processing allows the creation of no label tapes from standard label scratch volumes by changing the volume to master status and setting the initialize release action so that the tape is relabeled as a standard label tape when returned to scratch. DFSMSrmm also overrides the logical volume serial number generated by OPEN for NL output to scratch tapes, so that the correct

volume serial number is used for cataloging data sets. Nonstandard labels (NSL) can be used but only if the volume is not defined to DFSMSrmm. “How Does DFSMSrmm Validate Tape Mounts?” on page 17 describes how DFSMSrmm validates volume usage and restricts processing certain label types.

DFSMSrmm also supports the use of bypass label processing (BLP) for:

- Volumes that are either not defined to DFSMSrmm or that DFSMSrmm should ignore.
- Non-scratch (user and master status) volumes that are used for input processing, and only those user status volumes that are used for output processing. You can select this system option in the EDGRMMxx parmlib member using the OPTION BLP(RMM) command.
- Non-scratch (user and master status) volumes that are used for input processing, or user, master and scratch tapes that are used for output processing. You can select this option in the EDGRMMxx parmlib member using the OPTION BLP(NORMM) command.

When BLP is used with scratch tapes, DFSMSrmm changes the volume to master status. DFSMSrmm also sets the initialize volume release action to ensure that the volume has a valid standard label before it returns to scratch status. When a volume is written with BLP, DFSMSrmm can no longer perform the 44-character data set name check for the files on the volume.

For BLP output to a non-specific volume, DFSMSrmm does not check the file sequence number. For example, LABEL=(2,BLP) can be used. For non-BLP requests, DFSMSrmm restricts the use to LABEL=(1,label_type).

The data set information for files processed with BLP is only updated for output to first file on a volume. All other types of BLP requests change only the volume information such as the date last read and date last written. Also for BLP output requests, the data set information is only updated in the DFSMSrmm control data set when the data set is closed.

Data Sets

DFSMSrmm automatically records information about a data set when the data set is opened. For DFSMSrmm to automatically record information, data set recording must be on and the following must be true: the data set must be the first file on the volume, or you must process all files preceding it on the volume, or you must have previously defined the data sets on the volume to DFSMSrmm. Data class, management class, storage class, and storage group are included in the information that DFSMSrmm records for system-managed data sets.

You can manually add information about a data set if the volume on which the data set resides is already defined to DFSMSrmm and the volume has not been already processed as a result of open/close processing. There are some limitations on the information that you can change if it was originally recorded by DFSMSrmm during O/C/OEV processing.

DFSMSrmm supports generic data set names as filter criteria for searching the control data set, making it easier to create lists of resources.

Year 2000 Support

DFSMSrmm provides support for dates beyond the 20th century by ensuring that DFSMSrmm displays all dates using a 4-digit year. DFSMSrmm also allows you to specify dates using the 4-digit year. DFSMSrmm provides the same support for dates as DFSMSdfp.

Software Products

You can also define software products to DFSMSrmm and associate volumes with the products.

Owner Information

DFSMSrmm can help you keep track of volume owners, and provides functions for electronically notifying owners when their volumes are being considered for release. DFSMSrmm automatically records owner IDs as volumes are used. If you want DFSMSrmm to use owner information to automatically notify owners, you must manually define the owner's electronic address to DFSMSrmm. Additionally, the notify action must have been requested for the volume.

If a job is started which does not have a known RACF user ID, and RACF returns a user ID of '*', then DFSMSrmm uses the job's job name as the owner ID. DFSMSrmm automatically defines the owner ID.

DFSMSrmm also provides for notification to product owners when new volumes are added for a product.

DFSMSrmm keeps track of volume ownership. To delete a record for a user who still owns volumes, DFSMSrmm optionally allows you to transfer ownership of those volumes before deleting the owner record.

How Does DFSMSrmm Help You Create Reports?

You can obtain information and create reports using:

- The DFSMSrmm application programming interface
- DFSMSrmm ISPF dialog or RMM TSO commands
- EDGAUD and EDGRPTD report utilities
- DFSMSrmm EDGRRPTE exec shipped in SAMPLIB
- The DFSORT ICETOOL utility

Using the DFSMSrmm Application Programming Interface

You can use the DFSMSrmm application programming interface to obtain information about resources that are defined to DFSMSrmm. See *DFSMS/MVS DFSMSrmm Application Programming Interface* for information about how to use the DFSMSrmm application programming interface.

Using DFSMSrmm ISPF Dialog and RMM TSO Subcommands

You can search on-line using either the DFSMSrmm ISPF dialog or RMM TSO subcommands to create lists of resources and display information recorded in the DFSMSrmm control data set. Here are some examples:

- Operators can create lists of scratch volumes to be pulled for use.
- Tape librarians and system programmers can create lists of software products and the volumes on which they reside.
- General users can create lists of volumes they own, such as the example in Figure 3:

Volume	Owner	Rack	Assigned date	Expiration date	Location	Dsets	St	Act	Dest.
VOL600	AMYW01	RAC500	06/11/1994	11/11/1994	SHELF	0	UR	SI	
VOL601	AMYW01	RAC501	06/11/1994	11/11/1994	SHELF	0	UR	SI	
VOL603	AMYW01	RAC502	06/11/1994	11/11/1994	SHELF	0	UR	SI	

EDG3011I 3 ENTRIES LISTED

Figure 3. Example of a List of Volumes Owned by a Single User

With DFSMSrmm, you can use the RMM TSO SEARCH subcommands with the CLIST operand to create a data set of executable subcommands. For example, you can create subcommands to confirm volume movement for volumes identified during a SEARCHVOLUME request.

EDGAUD and EDGRPTD Report Utilities

You can create several types of reports using two DFSMSrmm report utilities. Use EDGRPTD to create movement and inventory reports and EDGAUD to create security and audit reports. EDGRPTD uses the DFSMSrmm extract data set as input. EDGAUD uses SMF records as input.

You can use the reports to:

- Identify volumes that should be moved between the removable media library and storage locations.
- Determine your volume inventory in the removable media library and storage locations.
- Identify volumes that are in transit or that should be marked as moved.
- Identify all accesses to volumes and changes to information recorded in the DFSMSrmm control data set.
- Separate volumes that are waiting to return to scratch from those that are private or have other release actions pending.

Using the DFSMSrmm EDGRRPTE Exec

DFSMSrmm provides standard reports and samples shipped in SAMPLIB. Use the EDGRRPTE exec to produce reports using the extract data set as input. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

Using the DFSORT ICETOOL Utility

You can use DFSORT or a similar program to generate a formatted report using the information in the extract data set created by the EDGHSKP utility. For example, you could produce an extract data set listing all volumes to be used on VM with information about volume owners. Then use the DFSORT ICETOOL utility to sort the information by volume and produce a report, complete with title and header information.

How Does DFSMSrmm Authorization and Security Work?

You can choose the authorization levels of users for all DFSMSrmm functions. DFSMSrmm uses MVS System Authorization Facility (SAF) for its authorization checking. You define DFSMSrmm resources to Resource Access Control Facility (RACF) for use during authorization checking. DFSMSrmm can create volume profiles, change them, and delete them on registration, expiration, or release of volumes. DFSMSrmm provides an access list you can use to set the access list in RACF. You can use the DFSMSrmm access list for authorization checking on non-RACF systems. Use the RMM LISTVOLUME subcommand or the DFSMSrmm ISPF dialog to display the DFSMSrmm access list. You can also view the access list in the volume records in the report extract data set.

DFSMSrmm provides automatic security classification through installation-specified criteria based on data set names. DFSMSrmm security includes:

- Audit trail of access and change of status through SMF. This audit trail produces information about RACF user ID, group, and job name.
- Operator confirmation required to use certain volumes.
- Erasure of data when a volume is released prior to the volume returning to scratch status.

DFSMSrmm provides the following ways of optionally keeping an audit trail for volumes defined to it:

- Control data set information
- SMF audit records
- RACF audit information

For more information about DFSMSrmm authorization and security, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

What Tape Usage Does DFSMSrmm Support?

DFSMSrmm supports tape usage for:

- Using BLP to read any private volume.
- Using BLP to write to any volume.
- Using nonstandard label volumes that are not defined to DFSMSrmm.
- Using both AL and SL scratch tapes.
- Using no label (NL) processing for any private volume, or performing NL output to standard label scratch volumes.
- Overwriting a volume that is in MASTER status. For volumes in MASTER status, DFSMSrmm allows the overwriting of data based on criteria set by the EDGRMMxx MASTEROVERWRITE operand.

- Overwriting a volume that is in USER status no matter what the data set name is.
- Using volumes not defined to DFSMSrmm.
- Using duplicate volumes if you use the EDGUX100 installation exit.
- Automatically labeling scratch tapes in an automated tape library dataserver.
- Reusing 36-track recorded scratch tapes on 18-track drives.
- Using multifile and multivolume data sets
- Recording and validating only the first file on the volume.

How Does DFSMSrmm Validate Tape Mounts?

DFSMSrmm performs validation for all data sets on a volume that have been recorded by DFSMSrmm when a data set is opened. DFSMSrmm validates tape volumes as follows:

- For specific tape requests, DFSMSrmm checks that the correct private volume is mounted.
- For non-specific tape requests, DFSMSrmm checks that a scratch volume is mounted and that the volume is from an acceptable rack or scratch pool.
- For volumes where DFSMSrmm has recorded the first file creation at open time, DFSMSrmm checks that the last 17 characters of the data set name from the tape volume HDR1 label match the first file data set name known to DFSMSrmm.
- For specific requests to overwrite data on a master volume, DFSMSrmm allows the overwriting of data based on criteria set by the EDGRMMxx MASTEROVERWRITE operand. DFSMSrmm checks that the data set name used matches the one DFSMSrmm has recorded. For generation data group data sets, DFSMSrmm removes the GDG suffix before checking the data set name.
- At OPEN time for the file being referenced, DFSMSrmm checks that the data set name used matches the one DFSMSrmm has recorded. If only the first file on a volume is being recorded, DFSMSrmm only validates the first file on the volume. If DFSMSrmm is reading a tape volume, only the last 17 characters of the data set name need to match the data set information in the tape header label.
- For scratch tapes mounted in a fully functional automated tape library dataserver, that have incorrect or missing VOL1 labels, DFSMSrmm ensures that both the external volume serial number and internal volume serial number, if one exists, are either defined as scratch to DFSMSrmm or are not defined to DFSMSrmm.

DFSMSrmm performs tape mount validation based on the DFSMSrmm running mode set in the parmlib member EDGRMMxx with the OPTION command and OPMODE operand.

Table 2. How the DFSMSrmm Running Mode Affects Tape Mount Validation

DFSMSrmm Running Mode	Tape Validation
Manual	DFSMSrmm does not validate volume usage.
Record-only	DFSMSrmm does not validate volume usage.
Warning	DFSMSrmm validates tape volume usage and issues warnings if errors are encountered. DFSMSrmm does not reject volume usage.
Protect	DFSMSrmm validates tape volume usage and rejects volume use under certain conditions.

Note: Use warning mode as you perform testing during conversion from another tape management system. DFSMSrmm validates your tape volumes but does not prevent their use. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information on setting options in the DFSMSrmm parmlib member.

Tape Volume Rejection by DFSMSrmm

In addition to validation, there are a number of reasons why a tape can be rejected when you are running DFSMSrmm in protect mode. Volumes can be rejected based on installation controlled parmlib options and volume definitions, or based on DFSMSrmm and system rules.

Rejects Caused by Installation Controls

With DFSMSrmm you can define conditions such as where volumes should reside, ranges of shelf locations that should not be used, and volumes that can only be used on certain systems. If you are running DFSMSrmm in protect mode the volume is rejected if:

- You have defined the REJECT option for a range of shelf locations, preventing a volume from being used on a particular system. The REJECT option is in parmlib member EDGRMMxx.
- The volume mounted for a scratch request has not been defined to DFSMSrmm.
- The volume mounted for a scratch request is not in a scratch pool associated with this system or does not match installation defined requirements.
- The volume mounted for a scratch request is from a scratch pool associated with another system.
- The volume is not to be used on MVS systems.
- The volume is not defined to DFSMSrmm and your REJECT options request that all non-defined volumes are to be rejected.
- A user asks for the use of a volume to be ignored, but is not authorized to do so for that volume.
- A volume in an automated tape library dataserer is not defined to DFSMSrmm, and cannot be defined to DFSMSrmm for some reason.

Rejects Caused by Validation Failure

DFSMSrmm performs validity checking on volumes when you read or write to them if DFSMSrmm is recording information about the data sets on the volumes. If you are running DFSMSrmm in protect mode the volume is rejected if:

- The wrong volume is mounted for a specific volume request.
- An attempt is made to use a specific scratch volume. In DFSMSrmm, when you want a specific volume, you must request a specific, non-scratch volume, and when you want a scratch volume, you must request a non-specific mount.
- A private volume (master or user) is mounted in response to a scratch request.
- The data set information for the first file on the volume that DFSMSrmm has recorded during OPEN/CLOSE/EOV processing does not match the information on the volume.
- An attempt is made to overwrite a data set on a master volume and the specified data set name does not exactly match the data set name that DFSMSrmm has recorded. You can control the overwriting of data sets on master volumes using

the EDGRMMxx MASTEROVERWRITE operand. If both the data set being written, and the data set DFSMSrmm has recorded are generation data group data sets, DFSMSrmm ignores the generation data group suffix when comparing the data set names.

If the volume is part of a multivolume sequence containing multiple data sets only the first volume, first file data set name is used for validation; for all other volumes the sequence of volumes is validated to prevent overwrite.

- An attempt is made to read a data set that DFSMSrmm has not recorded, and the volume information was previously recorded by DFSMSrmm.
- An attempt is made to automatically label a non-scratch volume in an automated tape library dataserver.

Rejects Caused by DFSMSrmm Rules

DFSMSrmm checks that volumes are labeled correctly and that the volume status and intended usage is acceptable to DFSMSrmm. If you are running DFSMSrmm in protect mode the volume is rejected if:

- An attempt is made to read a scratch volume.
- An attempt is made to read a volume obtained using the RMM GETVOLUME command and the volume has not yet been written to. GETVOLUME is used to request a scratch volume and assign it to an owner defined to DFSMSrmm.
- Bypass label processing (BLP) is being used to write to a scratch or master volume unless you requested BLP processing through EDGRMMxx in the parmlib.
- An attempt is made to read or write to a volume using nonstandard labels, and the volume is defined to DFSMSrmm.
- An attempt is made to overwrite standard labels on a master volume, and the user is not authorized to do so.
- An attempt is made to write standard labels on a master volume that has no labels, and the user is not authorized to do so.
- An attempt is made to overwrite standard labels on a scratch volume and the user is not authorized to do so.
- A scratch volume is requested for a nonstandard label request. In DFSMSrmm, scratch volumes must have standard labels.
- A volume is in an automated tape library dataserver and is defined to DFSMSrmm, but is defined as not having a standard label, or has different internal and external volume labels.
- An attempt is made to read or write to a volume that is waiting to be initialized.
- An attempt is made to read or write to a volume that is pending release.
- An attempt is made to write to a data set on the scratch volume other than the first.
- An attempt is made to write to a data set that was specified with a sequence number that is not the next in the sequence from the last file DFSMSrmm has recorded. This applies only if DFSMSrmm is recording information about all the data sets on the subject volume.

Who Can Use DFSMSrmm and How?

The following sections describe DFSMSrmm users and the tasks they can perform.

General User

General users need only limited access to DFSMSrmm functions. They might want to manage volumes they own and request information about resources defined to DFSMSrmm.

General users can use DFSMSrmm to:

- Manually request a scratch volume
- Change information about an owned volume or data set
- Update information about your owner ID
- Manually release an owned volume
- Create lists of resources and display information about most resources defined to DFSMSrmm

See “Chapter 2. Quick Reference for Basic DFSMSrmm Tasks” on page 23 for a quick description of these basic tasks.

Tape Librarian

Tape librarians can use DFSMSrmm to:

- Define new volumes
- Add shelf locations to the removable media library and to storage locations
- Add, change, and delete information about resources defined to DFSMSrmm
- Manually release any volume
- Confirm volume movements and actions
- Create lists of resources and display information about any resource defined to DFSMSrmm

Storage Administrator

Storage administrators can use DFSMSrmm to:

- Define retention and storage policies for data sets and volumes
- Change information about any volume they own, their owner ID, and any vital record specification
- Manually request a scratch volume
- Manually release a volume they own
- Delete a vital record specification
- Create lists of resources and display information about resources defined to DFSMSrmm

System Programmer

System programmers can use the DFSMSrmm support menu to:

- Display parmlib options and control data set control information
- Add, change, and delete information about any volume owner

- Manually request a volume, and manually release a volume they own
- Create a list of software products and display information about any resource defined to DFSMSrmm

Operator

Operators can use DFSMSrmm to:

- Fetch and mount tapes from specific pools or shelf locations, as specified in mount messages
- Manually erase and initialize tapes
- Manually request a scratch volume
- Manually release a volume they own
- Create lists of scratch tapes available for use

See “Chapter 11. DFSMSrmm Operator Procedures” on page 193 for a complete description of operator procedures.

Using DFSMSrmm

You define RACF profiles to establish the authorization scheme for using DFSMSrmm functions. The basic authorization scheme recognizes there are different types of users and each user type will request common DFSMSrmm functions. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information on how authorization is set up for each user type.

In DFSMSrmm, you can use either the DFSMSrmm ISPF dialog or the set of TSO subcommands to request DFSMSrmm functions. The RACF profiles control whether or not DFSMSrmm responds to requests for functions. If you request a function you are not authorized to use, your request will fail. For descriptions of the TSO subcommands available within DFSMSrmm, see “Chapter 12. Using RMM TSO Subcommands” on page 221.

DFSMSrmm offers menus in the DFSMSrmm ISPF dialog that are tailored specifically to a user group’s needs and level of access authorization. For example, only tape librarians are authorized to add software products to DFSMSrmm, so only the DFSMSrmm Librarian Menu includes an option to add software products. DFSMSrmm provides a specific menu for general users, storage administrators, tape librarians, and system programmers. It does not provide a menu for operators.

Figure 4 shows the menu designed for tape librarians:

```
Panel  Help
-----
EDGP@LIB                      DFSMSrmm Librarian Menu
Option ==>

0  OPTIONS    - Specify dialog options and defaults
1  VOLUME     - Display or Change volume information
2  ADDPP      - Add a Product
3  PRODUCTS   - Search for Products
4  OWNER      - Display or Change owner information
5  RACKS      - Manipulate Library Racks and Storage Location Bins
6  ADDVOL     - Add a volume to the library
7  SCRATCH    - Add SCRATCH volumes to the library
8  RELEASE    - Release volumes
9  CONFIRM    - Confirm librarian/operator actions
A  REQUEST    - Assign a user volume to an owner

Enter selected option or END command.  For more info., enter HELP or PF1.
```

Figure 4. DFSMSrmm Librarian Menu

Chapter 2. Quick Reference for Basic DFSMSrmm Tasks

This chapter is a quick reference for the basic tasks general users might need to perform using DFSMSrmm.

Using the DFSMSrmm Dialog

To start the DFSMSrmm dialog, select option R, the dialog entry for DFSMSrmm, from an ISMF selection menu and press ENTER.

The DFSMSrmm User Menu as shown in Figure 5 appears when the ISMF user mode is set to end user.

```
Panel  Help
-----
EDGP@USR                      DFSMSrmm User Menu
Option ==>

0  OPTIONS    - Specify dialog options and defaults
1  VOLUME     - Display list of volumes
2  DATA SET  - Display list of data sets
3  PRODUCTS   - Display list of Products
4  OWNER      - Display or Change owner information
5  REQUEST    - Request a new volume
6  RELEASE    - Release an owned volume

Enter selected option or END command.  For more info., enter HELP or PF1.

5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 5. DFSMSrmm User Menu

Select options 1 through 6 to request that DFSMSrmm perform functions against resources you own or to which you have access.

The User Menu is tailored to include the tasks you can perform as a general user. You have a level of access assigned to you; DFSMSrmm limits you to that level.

If your installation has customized an ISPF selection menu to include a dialog entry for DFSMSrmm, you can also start the dialog from there. Additionally, you can use the RMMISPF EXEC from an ISPF command or option line to start the DFSMSrmm dialog. See "Chapter 3. Getting Started with DFSMSrmm" on page 39 for more information.

Changing Dialog and Processing Options

Select option 0 on the User Menu to request the Dialog Options Menu, from which you can change dialog options, such as the sort order of lists and processing options, such as whether DFSMSrmm asks you to confirm delete and release requests.

Using Fast Path Commands

Issue fast path commands from the command or option line of any panel to request functions directly. This eliminates the need to return through previous panels.

For example, to see the Owner Details panel containing information about you as a volume owner, type OWNER DISPLAY and your owner ID, shown as DAVIDP in the following example:

```
==> OWNER DISPLAY DAVIDP
```

See “Issuing Fast Path Commands” on page 49 for a complete list of all fast path commands.

Requesting Help

Press PF1, or use the HELP command from any menu or function panel to view help panels containing information about the function, field-specific information, and examples.

Exiting the Dialog

To exit the dialog quickly, either:

- Type =X from any panel and press ENTER.
- Type X on the DFSMSrmm primary option menu and press ENTER.

You exit the dialog immediately.

To exit the dialog one panel at a time, use the END command or press the PF3 (END) function key from each panel to return to the primary option menu. If you used fast path commands to request several panels consecutively, you will return to those panels before exiting the dialog.

Getting Information about Your Resources

Table 3 describes how to request lists of resources that you own or over which you have ownership control, and how to request display panels containing detailed information about those resources:

Table 3. Requesting Lists and Display Panels from the User Menu

Type of Resource	Requesting Lists	Requesting a Display Panel
Volumes you own	<ol style="list-style-type: none">1. Select option 1 (VOLUME) on the User Menu and press ENTER.2. Enter any search criteria on the DFSMSrmm Volume Search panel and press ENTER.	<ol style="list-style-type: none">1. Request a list of volumes including the volume for which you want to view information.2. Type V in the line operator column (S) beside the volume serial number and press ENTER to see four panels of detailed information.

Table 3. Requesting Lists and Display Panels from the User Menu (continued)

Type of Resource	Requesting Lists	Requesting a Display Panel
Data sets on volumes you own	<ol style="list-style-type: none"> 1. Select option 2 (DATA SET) on the User Menu and press ENTER. 2. Enter any search criteria on the Data Set Search panel and press ENTER. 	<ol style="list-style-type: none"> 1. Request a list of data sets including the data set for which you want to view information. 2. Type I in the line operator column (S) beside the data set name and press ENTER to see detailed information about the data set.
Software products	<ol style="list-style-type: none"> 1. Select option 3 (PRODUCTS) on the User Menu and press ENTER. 2. Enter any search criteria on the Product Search panel and press ENTER. 	<ol style="list-style-type: none"> 1. Request a list of software products including the one for which you want to view information. 2. Type P in the line operator column (S) beside the software product number and press ENTER to see detailed information about the software product.
Your owner ID	(Does not apply)	<ol style="list-style-type: none"> 1. Select option 4 (OWNER) on the User Menu and press ENTER. 2. Select option 1 (DISPLAY) and supply your owner ID.

You can use line operators from lists to request functions. Press PF1 or use the HELP command from a displayed list to view help panels containing information about individual line operators.

Sorting Lists

You can change the default sort order DFSMSrmm uses to sort the lists you request by using sort options panels. To sort a list while looking at it, use the SORT command. See “How to Sort a List” on page 176 for more information.

Changing the Default Sort Order

On the sort options panels, you give the table field names and the order (ascending or descending) in which DFSMSrmm should sort the list. Table field names correspond to data columns on lists, each containing particular information recorded by DFSMSrmm. For example, a list of volumes contains several data columns: DFSMSrmm displays a volume serial number, an assigned date, a release date, a location, the number of data sets on the volume, a shelf location in the removable media library, and a media name for each volume.

To change the default sort order for a list:

1. Select option 0 (OPTIONS) on the User Menu and press ENTER. DFSMSrmm displays the Dialog Options Menu.
2. Select option 2 (SORT) on the Dialog Options Menu and press ENTER. DFSMSrmm displays the Dialog Sort Options Menu.

3. Select an option corresponding to the type of list you want and press ENTER. DFSMSRmm displays a sort options panel listing the sort options corresponding to the specific table field names on the list you selected.

For example, select option 4 (VOLUME) to see the Volume List Sort Options panel, as Figure 6 shows:

Panel Help

EDGP@STV

DFSMSRMM VOLUME LIST SORT OPTIONS

Command ==>

Enter the relative sort priority (1 to 18) and direction (A or D):

Table field name	Priority	Direction	Table field name	Priority	Direction
Volume Serial			Rack number		
Owner			Media name		
Assigned date			Home location		
Expiration date			Media type		
Status			Label		
Location			Recorded format		
Destination			Compaction		
Transit			Attributes		
Data Sets			Action		

Enter END command to save changes, or CANCEL to end without saving.

Figure 6. DFSMSRmm Volume List Sort Options Panel

4. Enter the sort priorities (1 to 18) and sort directions (ascending or descending) of your choice. Press PF1 or use the HELP command to see help panels containing more information.
5. Press ENTER to process your sort options.

Changing Sort Order While Looking at a List

Use the SORT command from the option line of any list to change the list's sort order while you look at it. The values you use change the default sort order for your session.

Type SORT followed by one or more parameter pairs, where each parameter indicates a direction (ascending or descending) and a data column name or abbreviation. For example, to sort a volume list by media name in ascending order, type:

```
==> SORT ASCENDING MEDIANAME
```

You can use several parameter pairs with the SORT command. For example, to sort your volume list first by media name in ascending order and then by rack number in descending order, type:

```
==> SORT ASCENDING MEDIANAME DESCENDING RACK
```

Printing Display Panels and Lists

To print a displayed list or a details panel, use the ISPF commands PRINT and PRINT-HI.

Use the PRINT command to print panels a screen at a time and save the output in your ISPF list file. If the displayed information exceeds one screen, you must scroll to the next screen of entries and issue the PRINT command for each screen of entries.

Use the PRINT-HI command to print panels so that any highlighted characters on a panel appear in bold print.

You can assign either PRINT or PRINT-HI to a PF key. See *OS/390 ISPF User's Guide, SC28-1239* for more information.

Requesting Lists and Resource Details in Batch

You can request lists of resources and details about those resources, in batch:

1. Select option 0 (OPTIONS) from the User Menu and press ENTER.
2. Select option 1 (USER) from the Dialog Options Menu and press ENTER.
3. Specify B in the Processing options field:

Processing options ==> B

DFSMSrmm saves your commands in a data set instead of processing them interactively.

4. Go to a list panel of your choice and enter the search criteria to request your list.
5. Press ENTER. DFSMSrmm returns a message indicating that your command has been saved.
6. Press PF4 or use the RETURN command to return to the DFSMSrmm primary option menu.
7. Press PF3 (END) from the primary option menu to see the Exit Menu.
8. Select option S and enter the job statement information from the Exit Menu to submit your batch job.

To request resource details in batch:

1. Request a list of resources that includes the resource for which you want details.
2. Type SAVE ON at the option line of the list panel.
3. Enter the line operator that specifies a display action against the entry of your choice.

For example, to see all details for a volume, type V beside the volume serial number.

4. Press ENTER. DFSMSrmm returns a message indicating that your command has been saved.
5. Press PF4 or use the RETURN command to return to the DFSMSrmm primary option menu.
6. Press PF3 (END) from the primary option menu to see the Exit Menu.
7. Select option S and enter the job statement information from the Exit Menu to submit your batch job.

Requesting a Scratch Volume

Usually, you receive a scratch volume automatically when you run a batch job that requests a non-specific tape mount. When you write data to a scratch volume, DFSMSrmm changes the volume status to a master volume. For master volumes, DFSMSrmm allows the overwriting of data based on criteria set by the EDGRMMxx MASTEROVERWRITE operand. We recommend that you request scratch volumes in this way.

To manually request a scratch volume without running a batch job:

1. Select Option 5 (REQUEST) on the User Menu and press ENTER.
DFSMSrmm displays the Request a Volume panel, as Figure 7 shows:

Panel Help	

EDGPT500 DFSMSrmm Request a Volume	
Command ==>	
Owner MAZOWN44	Pool
	or
Owner Access	Media Name
Retention Period . .	Expiration Date . . . (YYYY/DDD)
Description	
	Release Actions
Security Name . . .	Return to SCRATCH Pool . . YES
	Replace volume NO
MVS Use	Return to owner NO
VM Use	Initialize volume NO
	Erase Volume NO
Location	Notify Owner YES
Volume Access List:	Access
User . . User . . User . . User . .	
User . . User . . User . . User . .	
User . . User . . User . . User . .	
Press ENTER to ADD the volume, or END command to CANCEL.	

Figure 7. DFSMSrmm Request a Volume Panel

2. Include any optional information. Press PF1 or use the HELP command for field-specific information.

You are assigned as the volume owner by default, unless you supply a different owner ID.

Specify a retention period or an expiration date for the volume, unless you want to use your installation's default retention period.

If you want a volume to be chosen from a particular pool, use a pool ID. Unless you give a pool ID, the volume is chosen from a default scratch pool. To display the pool IDs defined for your location, type CONTROL VLPOOLS from the command line.

Select one or more release actions to indicate what should be done with the volume when it becomes eligible for release. For example, enter YES in the NOTIFY OWNER field if you want to be notified when the volume becomes eligible for release.

3. Press ENTER.

DFSMSrmm assigns the volume and changes its status to user volume, meaning the volume can be overwritten at any time by a user authorized to use the volume.

Releasing a Volume Manually

You should not have to release volumes manually, unless you no longer want them. DFSMSrmm automatically determines when a volume is eligible for release, and schedules any release actions that have been requested for it.

You can manually release a volume you own anytime before the expiration date is reached. This release will override any retention policies defined by your storage administrator for the volume, or for any data sets on the volume. When you manually release a volume, it becomes eligible for return to scratch status and possible use by others. DFSMSrmm retains information about the volume and any data sets on the volume until the volume is reused or, if the initialize and erase actions have also been set for the volume, when those actions are confirmed as complete.

To release one or more volumes you own:

1. Select option 6 (RELEASE) from the User Menu and press ENTER.
DFSMSrmm displays the Release Volumes panel, as Figure 8 shows:

Panel Help

EDGPT420 DFSMSrmm Release Volumes

Command ==>>

Volume . . VOL000

Limit . . . Limit search to first nnnn volumes

The following line commands will be available if a list is displayed:

R - Release volume

V - Display volume information

Press ENTER to SEARCH, or END command to CANCEL.

Figure 8. DFSMSrmm Release Volumes Panel

2. Do one of the following:
 - Leave the volume serial field blank to request a list of all volumes you own. Enter a limit if you want to see a specific number of entries in the list.
 - Supply the volume serial number of the specific volume you want to release.
3. Press ENTER to release the owned volume or to request a list.

- If you asked for a particular volume serial number and you also requested you confirm release requests before DFSMSrmm releases the volumes, DFSMSrmm displays a Confirm Volume Release panel, as Figure 9 shows. Press ENTER to confirm that you want to release the volume. Press PF3 to cancel the request.

```

Panel  Help
-----
EDGPT410          DFSMSrmm Confirm Volume Release
Command ==>

Volume . . . . . : VOL000          Location . . . . . : DC
Volume type . . . : PHYSICAL       In container . . . . :
Media name . . . . : 3480          Rack number . . . . . : C00000
                                   Expiration date . . . . :

Status . . . . . : SCRATCH         Original expiration date :
Description . . . :
Data set name . . :

Media type . . . . : *              Release actions:
Label . . . . . : SL               Return to SCRATCH pool : YES
Density . . . . . :                Replace volume . . . . : NO
Recording format . : *              Return to owner . . . : NO
Compaction . . . . : *             Initialize volume . . : NO
Attributes . . . . : NONE           Erase volume . . . . : NO
Availability . . . :                Notify owner . . . . : NO
                                   Expiry date ignore . . : NO
                                   Scratch immediate . . : NO

Press ENTER to RELEASE volume, or END command to CANCEL.

```

Figure 9. DFSMSrmm Confirm Volume Release Panel

- If you requested a list, DFSMSrmm displays a list of volumes you own, similar to the list shown in Figure 10. Type R in the line operator column (S) beside the volume serial number for each volume you want to release and press ENTER.

```

Panel  Help  Scroll
-----
EDGPT020          DFSMSrmm Volumes (Page 1 of 2)          Row 1 to 8 of 8
Command ==>                                              Scroll ==> PAGE

The following line commands are valid: C,I,O,R,E,L and V
Use the RIGHT command to view other data columns

Volume   Assigned  Expiration  Dest-  Tra-  Data
S serial Owner   date       date      Status Location ination nsit sets
-----
R A05000 RMMUSER 1999/012   1999/017  MASTER  SHELF          N    0
  A05001 RMMUSER 1999/012   1999/017  MASTER  SHELF          N    0
  A05002 RMMUSER 1999/012   1999/017  MASTER  SHELF          N    0
  A05003 RMMUSER 1999/012   1999/017  MASTER  SHELF          N    0
  A05004 RMMUSER 1999/012   1999/017  MASTER  SHELF          N    0
  A05005 RMMUSER 1999/012   1999/017  USER    SHELF          N    0
  A05006 RMMUSER 1999/012   1999/017  USER    SHELF          N    0
  A05007 RMMUSER 1999/012   1999/017  USER    SHELF          N    0
***** Bottom of data *****

```

Figure 10. Example of Volume List

Changing Owner Information

To change information in your own owner record:

1. Select Option 4 (OWNER) on the User Menu. DFSMSrmm displays the Owner Menu.
2. Select Option 3 (CHANGE). Ensure that your owner ID is entered in the OWNER ID field, then press ENTER.

DFSMSrmm displays a Change Owner Details panel that defines information for your owner ID, such as the panel shown in Figure 11:

```
Panel  Help
-----
EDGP0300          DFSMSrmm Change Owner Details - MAZOWN44
Command ==>

Surname . . . . SEYFERT
Forenames . . . CLEMENS

Department . . . 4193

Address:
Line 1 . . . . 123 Rock Road
Line 2 . . . . San Jose
Line 3 . . . . CA

Telephone:
Internal . . . 123-4567      External . . . 98765-5678

Electronic Mail:
Userid . . . . USER01      Node . . . . NODE01
Press ENTER to CHANGE Owner Details, or END command to CANCEL.
```

Figure 11. DFSMSrmm Change Owner Details Panel

3. Change any information on the panel, or add any information missing from your record. For example, if you want to be notified when a volume you own is about to be released, your owner record must include both a valid user ID and node. A user ID and node can each be up to eight characters.
Press PF1 or use the HELP command for field-specific information.
4. Press ENTER to process your changes.

Changing Volume Information

Each time a data set on a volume is opened or closed, DFSMSrmm automatically records some volume details. There are some restrictions on changing the information recorded automatically by DFSMSrmm. See “CHANGEVOLUME: Changing Volume Information” on page 282 for a list of the information you can change.

To manually change information about a volume you own that is already defined to DFSMSrmm:

1. Select Option 1 (VOLUME) on the User Menu and press ENTER.

DFSMSrmm displays the DFSMSrmm Volume Search panel, as Figure 12 shows.⁵

Panel
Help
Scroll

EDGPT010
DFSMSrmm Volume Search

Command ==>

Volume
Owner
Media name . .
Pool
Status
Since
Retention . . .
Limit
Home
Location . . .
In container
Volume type
Media Type . .
Label
Current version
Required version
Density
Format
Compaction . .
Attributes . .
Destination . .
Move mode . . .
Intransit . . .

More: +

May be generic. Leave blank for all volumes.
Owned by a specific user. Default is your userid
Limit to a single media name or to a particular pool
Select volume status. Default is ALL.
Volumes assigned since YYYY/DDD
Volumes retained up to YYYY/DDD
Limit search to first nnnn volumes
Limit to volumes with this home location name
Limit to volumes in a single location
Physical stacked volser
(LOGICAL or PHYSICAL)
Tape media type (*, CST, ECCST, HPCT or EHPCT)
Tape label standard (for example SL)
Label version number(for example 3)
Label version number(for example 4)
Tape recording density
Tape recording format (*, 18TRACK, 36TRACK or 128TRACK)
Limit to volumes containing compacted data
Tape special attributes (NONE or RDCOMPAT)
Limit by destination
Limit by move mode (AUTO or MANUAL)
Limit to volumes which are moving (YES or NO)

The following line commands will be available when the list is displayed:
C - Change volume information
I - Data set information
O - Owner information
V - Volume information
E - Eject volume
L - List volume chain
R - Release volume

Enter SCROLL commands for more search values,ENTER to SEARCH or END to CANCEL.

Figure 12. DFSMSrmm Volume Search Panel

2. Enter any of the following in the Volume field:
 - A blank or * for a list of all volumes you own
 - A generic volume serial number for a list of volumes you own with similar volume serial numbers
 - A specific volume serial number for a list containing only one volume you own
3. Use the Limit field to increase the number of volumes DFSMSrmm returns in the list if more than 10 volumes are to be returned. This ensures that the volume that you want to change is included in the list. You can also tailor the search criteria. For example, you can request that DFSMSrmm only list volumes assigned to you since a specific date.

⁵ Import/Export support is available with APAR OW36342 or OW36343.

Press PF1 or type HELP at the command line for field-specific help.

4. Press ENTER.

DFSMSrmm returns a list of volumes in the Volume List Panel.

5. Find the volume that you want to change, and type C in the line operator column (S), beside the volume serial number.

6. Press ENTER.

DFSMSrmm displays the Change Volume Details panel, containing information about the volume you specified. For example, DFSMSrmm displays a panel such as the one shown in Figure 13:

Panel

Help

EDGPT310

DFSMSrmm Change Volume Details -

Command ==>

Volume : A00001

Pool

Media name : 3480

Storage group . . .

Move mode AUTO

Volume type PHYSICAL

Retention period . .

Status SCRATCH

Description

Data set name . . . 'SHELF.REPORT.DS001'

Media type *

Label SL

Current version

Required version

Density *

Recording format . .

Compaction

Attributes

Availability

Owner

Assigned date . . . 1998/253

Last read date . . .

Security name . . .

Classification . . .

Account number . . .

Jobname

Loan location . . .

Previous volume . .

Volume access list :

User

User

User

User

User

User

More: +

or

Rack A00001

Location name . . .

Destination

Bin number

In container

Expiration date . . 1999/015

Initialize NO

Release actions:

Return to SCRATCH pool . .

Replace volume

Return to owner

Initialize volume

Erase volume

Notify owner

Expiry date ignore . . . : NO

Scratch immediate . . . : NO

Owner access

Assigned time 00:27:17

Last write date

MVS use YES

VM use NO

Last changed by : *HKP

Next volume

Access

User

User

User

User

User

User

Enter SCROLL commands for more volume information, END command to CANCEL.

Figure 13. DFSMSrmm Change Volume Details Panel

7. Scroll down to change or add missing information to any of the fields on the panel, and press ENTER.

Changing Release Date for a Volume

DFSMSrmm uses the latest expiration date of all the data sets on the volume to determine the date a volume can be released. You can change this date for a volume you own without writing to the volume again. Do this any time after the volume has been defined to DFSMSrmm and before the release date is reached.

To change the release date for a volume you own, change either the volume's retention period or its expiration date on the DFSMSrmm Change Volume Details panel as shown in Figure 13 on page 33. The new release date you give cannot exceed the maximum retention period set by your installation.

For example, if you want to retain this volume for another six months, you can change the expiration date shown in the example from 1999/015 to 1996/195.

If you want the volume to be considered for release immediately, you can release it manually. See "Releasing a Volume Manually" on page 29 for more information.

Setting Release Actions for a Volume

You can request actions to be taken when a volume you own becomes eligible for release. For example, you can ask to be notified when a volume you own is about to be released, or that it is returned to you upon release. If you do not request any release actions for a volume, DFSMSrmm automatically returns it to scratch status upon release. You can use any one of the following mutually exclusive actions:

Return to scratch

To request that the volume be returned to scratch status.

Replace volume

To request that the volume be replaced with a new volume and returned to scratch status.

Return to owner

To request that the volume be returned to you as its owner.

In addition, you can request any or all of the following actions:

Initialize volume

To request that the volume be initialized. If you select this, the volume is not available for reuse until it has been initialized.

Erase volume

To request that the volume be erased. If you select this, the volume is not available for reuse until it has been erased.

Notify owner

To request that you be notified when the volume is released. If you select this, your owner record must include a valid user ID and node.

You can request up to four release actions for a volume. The four actions would consist of one from the mutually exclusive group (return to scratch, replace volume, or return to owner), and all three remaining actions (initialize volume, erase volume, notify owner).

To set a release action, change the value in the Release Actions field on page one of the Change Volume panel. Specify YES in the entry field opposite the release actions of your choice. Specify NO if you do not want a release action.

Changing Data Set Information

DFSMSRmm automatically records some data set details each time a data set is opened or closed. DFSMSRmm puts restrictions on the automatically recorded information that you can change. You can change the security level recorded for a data set without restriction. To change other fields, DFSMSRmm requires that you have access to STGADMIN.EDG security resources. To change the date that the data set was last read or written to, you must have CONTROL access to STGADMIN.EDG.MASTER security resource. To change other data set details that were recorded by DFSMSRmm during O/C/EOV processing, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE. See “CHANGEDATASET: Changing Data Set Information” on page 270 for information on the data set details that you can change.

To change information recorded by DFSMSRmm for a particular data set on a volume you own:

1. Select Option 2 (DATA SET) on the User Menu and press ENTER. DFSMSRmm displays the Data Set Search panel, as Figure 14 shows:

Panel Help

EDGPD010 DFSMSRmm Data Set Search

Command ==>

Enter fully qualified or partial data set name and job name:

Data set name DATA.*

Job name

Enter optional parameters to qualify search

Owner Owner of volumes (Default is your userid)

Volume Serial OR Volume serial

Create Date Created since YYYY/DDD

Physical file seq . . Leave blank for all

Limit Limit search to first nnnn data dets

The following line commands will be available when the list is displayed:

I - Data set information V - Volume information

O - Owner information C - Change data set information

D - Delete data set information

L - List multi-volume data set chain

Figure 14. DFSMSRmm Data Set Search Panel

You might also want to tailor the search criteria. For example, if you want a list of only data sets that match a data set name, enter a data set name. Use the Limit field if you want DFSMSRmm to return more than 10 data sets in the list to ensure that the data set you want to change is included in the list.

Press ENTER to request a list of data sets on volumes you own. DFSMSRmm returns a list of data sets, such as the list shown in Figure 15 on page 36:

Panel Help

EDGPD020 DFSMSrmm Data sets Row 1 to 8 of 8

Command ==> Scroll ==> PAGE

The following line commands are valid: C, D, I, L, O, V.

		Volume	File	Create
S	Data set name	Serial Owner	Seq	Date
	DATA.SET.EIGHT	VOL002 OWN000	2	1998/030
C	DATA.SET.FIVE	VOL001 OWN000	1	1998/090
	DATA.SET.FOUR	VOL000 OWN000	4	1998/080
	DATA.SET.ONE	VOL000 OWN000	1	1998/050
	DATA.SET.SEVEN	VOL002 OWN000	1	1998/070
	DATA.SET.SIX	VOL001 OWN000	2	1998/030
	DATA.SET.THREE	VOL000 OWN000	3	1998/040
	DATA.SET.TWO	VOL000 OWN000	2	1998/060

***** Bottom of data *****

Figure 15. Data Set List Example

- Find the data set you want to change and type C in the line operator column (S), beside the data set name. Then press ENTER.
DFSMSrmm displays a Change Data Set Details panel containing information defined for the data set. For example, DFSMSrmm displays a panel such as the one shown in Figure 16 on page 37:

```

Panel  Help
-----
EDGPD310          DFSMSrmm Change Data Set Details
Command ===>

Data set name . . . : 'DATA.SET.FIVE'
Volume serial . . . : VOL000      Physical file sequence number . . . : 1
Owner . . . . . : OWN000      Data set sequence number . . . . . 0
                                           More:      +

Job name . . . . .
Step name . . . . .      Record format . . . . .
DD name . . . . .      Block size . . . . . 0
Create date . . . . 1998/080   YYYY/DDD      Logical record length  0
Create time . . . . 00:31:18   Block count . . . . . 0
System id . . . . .      Unit address . . . . .

Date last read . . .      VRS management value
Date last written . . .   Management class . . .
                          Data class . . . . . :
Retention date . . .      Storage class . . . . . :
VRS retained . . . : NO    Storage group . . . . . :

Security name . . . VTAPE
Classification . . : VTAPE CLASS TO FORCE VOL ERASE

Primary VRS details:
  VRS name . . . :
  Job name . . . :      VRS type . . . . . :
  Subchain name :      Subchain start date :

Secondary VRS details:
  Value or class :
  Job name . . . :
  Subchain name :      Subchain start date :

Catalog status . . : UNKNOWN
Abend while open . . : NO

Press ENTER to CHANGE the data set name details, or END command to CANCEL.

```

Figure 16. DFSMSrmm Change Data Set Details Panel

3. Enter information in the panel to change information. There are some restrictions on the changes you can make. For example, if DFSMSrmm automatically recorded information about the data set when the data set was opened, you can only change the data set's security level, date last read or date last written to.

Press PF1 or use the HELP command for field-specific help for information on the fields you can change.
4. Press ENTER to request your changes.

Requesting Notification of a Volume's Release

If you want to be notified when a volume you own is being considered for release, use the Change Volume panel. You must set the Notify Owner release action to YES for that volume anytime before the volume is to be released. See “Setting Release Actions for a Volume” on page 34 for more information.

You must have a valid user ID and node defined to DFSMSrmm in your owner record before DFSMSrmm can notify you of the volume's release. Use the Change Owner Details panel to add missing information or to change an incorrect user ID and node. See "Changing Owner Information" on page 31 for more information.

Requesting Return of a Volume after Release

If you want a volume you own to be returned to you when it becomes eligible for release, set the Return to Owner release action to YES for that volume. Use the Change Volume panel to do this anytime before the volume is due to be released. See “Setting Release Actions for a Volume” on page 34 for more information.

Requesting Special Volume and Data Set Actions

Contact your storage administrator if you have unique requirements for your data sets and volumes, such as retaining and storing multiple cycles of a data set, or grouping your volumes together in the removable media library.

Retaining Data Sets and Volumes

Your storage administrator can define retention policies, known as vital record specifications, to retain data sets or volumes you own based on your needs. These policies override any expiration date associated with the data set or volume.

If you want to retain a data set on a volume you own, provide the following information to your storage administrator:

- The data set name and whether or not it is part of a GDG
- The job name that created the data set if you want to retain or move volumes based on job name
- That you want to retain the data set by number of days or cycles
- How many days or how many cycles of the data set to retain
- How long you want the retention policy to be in effect
- Whether the data set or volume has any off-site storage requirements

If you want to retain a volume you own, provide the following information to your storage administrator:

- The volume serial number
- How long you want the retention policy to be in effect
- Where you want the volume to be stored

You can use a data set name mask or volume serial number on a retention policy to retain several data sets or volumes. If you use a generic volume serial number, you must also indicate the total number of volumes to be retained by the policy.

Grouping Your Volumes

If you need to keep your volumes grouped together in a pool in the removable media library, ask your storage administrator to define a pool for you. Provide your storage administrator with the following information:

- The number of existing volumes and volumes to be added in the future that the pool must accommodate.
- The status of volumes to be stored in the pool. For example, are the volumes read-only, or can they be used to satisfy scratch requests once the data on them expires?

Chapter 3. Getting Started with DFSMSrmm

This chapter describes how to start and stop a DFSMSrmm session through the ISPF dialog. It also shows some of the modifications you can make to the dialog during your session, and how to use menus and fast path commands to request functions.

Before you can start a DFSMSrmm session, you must have the correct libraries available. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on loading libraries for panels, messages, skeletons and execs.

Starting the Dialog

Start your DFSMSrmm session from an ISMF selection menu, from an ISPF primary option panel, or by using the RMMISPF EXEC.

When you start your session, the DFSMSrmm primary option menu as shown in Figure 17 is the first menu you see. You can change the first menu DFSMSrmm displays if your DFSMSrmm dialog invocation is automatically set in your logon procedure, if you select an ISMF option, or use the RMMISPF EXEC with an operand.

```
Panel  Help
-----
EDG@PRIM          REMOVABLE MEDIA MANAGER (DFSMSrmm)
Option ==>>

0  OPTIONS      - Specify dialog options and defaults
1  USER        - General user facilities
2  LIBRARIAN    - Librarian functions
3  ADMINISTRATOR - Administrator functions
4  SUPPORT      - System support facilities
5  COMMANDS     - Full DFSMSrmm structured dialog
6  LOCAL        - Installation defined dialog
X  EXIT         - Exit DFSMSrmm Dialog

Enter selected option or END command.  For more info., enter HELP or PF1.

5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 17. DFSMSrmm Primary Option Menu

Starting DFSMSrmm from an ISMF Selection Menu

You can select an option under ISMF to define a user mode to control which panel you are shown first.

Select option R, the dialog entry for DFSMSrmm, from an ISMF selection menu and press ENTER. If you select storage administrator mode, DFSMSrmm displays the DFSMSrmm primary option menu as shown in Figure 17. If you select end user

mode, DFSMSrmm displays the DFSMSrmm User Menu as shown in Figure 5 on page 23 . See *DFSMS/MVS Using ISMF* for additional information.

Starting DFSMSrmm from an ISPF Primary Option Panel

If your installation has added a dialog selection to one of your existing ISPF primary option panels, you can request the dialog from there. For information about adding a DFSMSrmm selection to the ISPF primary option menu, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* .

Using the RMMISPF EXEC

You can request the RMM dialog directly from TSO, or from the ISPF command line using the RMMISPF EXEC:

At the TSO READY prompt, enter:

RMMISPF

From the ISPF command line, enter:

TSO RMMISPF

Press ENTER.

Figure 18 shows the operands you can use with the RMMISPF EXEC to bypass the DFSMSrmm primary option menu:

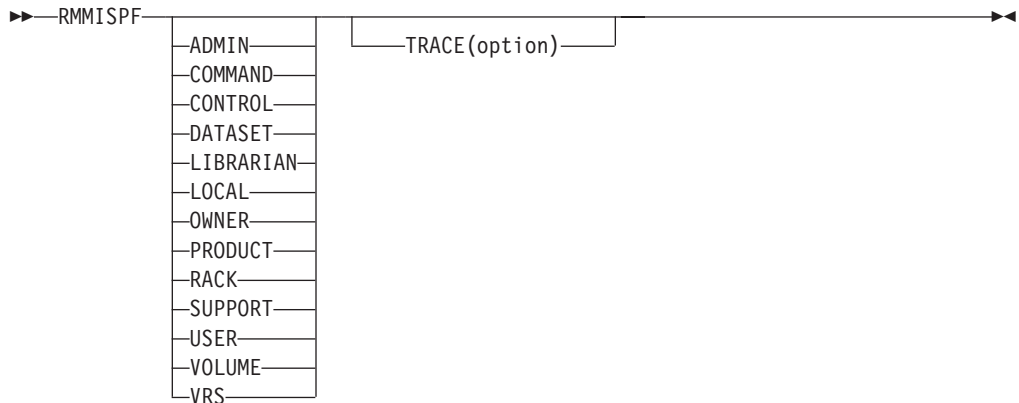


Figure 18. RMMISPF Exec Syntax Diagram

Each of the operands, except for TRACE, represents a menu from which you can request functions. For example, if you request the following from the TSO environment:

RMMISPF LIBRARIAN

DFSMSrmm displays the Librarian Menu.

Use the TRACE(option) operand to diagnose problems in any of the REXX execs supplied in the dialog. You can specify ALL, OFF, or the name of one or more REXX execs as options for TRACE. For more information on using the TRACE

operand see *DFSMS/MVS DFSMSrmm Diagnosis Guide*. For information on bypassing the DFSMSrmm primary option menu, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

Setting Dialog Options

Use the Dialog Options Menu to change any of the following dialog options:

- Processing options
- List sorting options

To display the Dialog Options Menu, do one of the following:

- Select Option 0 (OPTIONS) from any menu and press ENTER.
- or
- Type OPTIONS on the command or option line of any DFSMSrmm panel and press ENTER.

DFSMSrmm displays the DFSMSrmm Dialog Option Menu as shown in Figure 19:

```
Panel  Help
-----
EDGP@OPT          DFSMSrmm Dialog Options Menu
Option ==>

1  USER          - Specify processing options
2  SORT           - Specify list sort options

Enter selected option or END command.  For more info., enter HELP or PF1.

5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 19. DFSMSrmm Dialog Options Menu

Changing Processing Options

You can change DFSMSrmm processing options using the Dialog User Options panel shown in Figure 20 on page 42.

1. To change processing options, select Option 1 (USER) on the Dialog Options Menu and press ENTER. Figure 20 on page 42 shows the Dialog User Options panel DFSMSrmm displays:

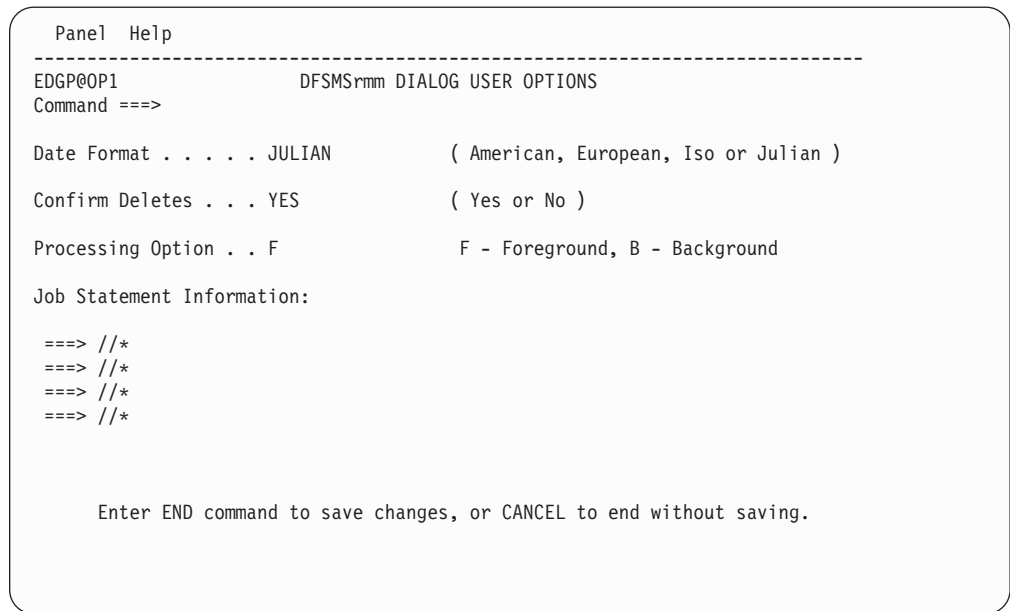


Figure 20. DFSMSrmm Dialog User Options Panel

2. You can change:

- The date format used with lists and panels
- When DFSMSrmm displays delete request confirmation panels
- Whether DFSMSrmm processes your requests interactively or saves them in a data set to be processed when you exit the dialog.

All of the information you enter on this panel is optional. Use the END command to save any changes you make.

Setting the Date Format

Choose from the options listed in Table 4 to set the date format DFSMSrmm uses when it displays lists or panels:

Table 4. Date Format Options

Language	Format	Example
American	mm/dd/yyyy	12/15/1994
European	dd/mm/yyyy	15/12/1994
Iso	yyyy/mm/dd	1994/12/15
Julian	yyyy/ddd	1994/349

The date format is initially set to JULIAN. When you change this initial setting, DFSMSrmm stores the new date format and remembers it across sessions.

You can change the date format at any time by returning to the Dialog User Options panel or by using the DATE command. Use the DATE command from the command or option line of any panel to bypass the Dialog User Options panel. Use DATE with any of the following parameters: AMERICAN, EUROPEAN, ISO, or JULIAN. For example,

```
==> DATE EUROPEAN
```

causes DFSMSrmm to set the date format to dd/mm/yyyy. If you use DATE with no parameters, DFSMSrmm displays the current setting for date format. DFSMSrmm always displays dates using 4 characters for the year, so that 21st century dates can be supported. For example, 15th March 95 is 15/03/1995 in European format.

Requesting to Confirm Delete and Release Requests

You can confirm delete and release actions before DFSMSrmm performs them. For example, if you delete a data set on a volume, using the Delete Data set panel, and you requested the confirm option for your session, DFSMSrmm displays the Confirm Delete Data Set panel. This panel shows information about the data set you chose and asks for confirmation that you want this data set deleted from DFSMSrmm.

Put YES in the Confirm Deletes field of the Dialog User Options panel if you want to confirm all delete and release actions before DFSMSrmm performs them. Put NO if you do not want to confirm any actions.

The confirm delete option is initially set to YES. Once you change this initial setting, DFSMSrmm stores the new setting and remembers it across sessions.

You can change the confirm delete option at any time during your session by returning to the Dialog User Options panel or by using the CONFIRM command. Use the CONFIRM command from the command or option line of any panel to bypass the Dialog User Options panel.

Use CONFIRM with either ON or OFF. Enter

```
====> CONFIRM ON
```

and DFSMSrmm prompts you to confirm a specific delete or release action.

You can change this option repeatedly during your DFSMSrmm ISPF session. See “Deleting Software Product Information” on page 71 for more information on the individual confirm delete panels.

Selecting Command Processing

You can select how DFSMSrmm is to process your requests for functions. Choose either:

Foreground

DFSMSrmm processes your requests interactively

Background

DFSMSrmm saves your requests in a data set

The default is foreground processing.

It is useful to select background processing if you are requesting functions at the same time that inventory management is running. You can specify background processing so that DFSMSrmm builds and saves your commands for processing at a later time.

You can change command processing at any time during your session by returning to the Dialog User Options panel or by using the SAVE command. Use the SAVE command from the command or option line of any panel to bypass the Dialog User Options panel. Use SAVE with either ON or OFF. For example,

====> SAVE ON

causes DFSMSrmm to start saving your requests. Specify SAVE OFF to stop further requests from being saved. You can turn the SAVE option on and off repeatedly in your session.

DFSMSrmm remembers the command processing option you select for only one session and reverts to foreground processing when you end your session.

Note: If you use background processing, DFSMSrmm saves only *intended* requests. DFSMSrmm does not save other commands that are needed to enable an intended request to be processed successfully. For example, if you request to change volume information, a DFSMSrmm TSO CHANGEVOLUME subcommand is built. First, DFSMSrmm issues a LISTVOLUME subcommand to display the current volume details. You change the details you want, and then DFSMSrmm builds and saves a CHANGEVOLUME subcommand.

If you select background processing and DFSMSrmm saves at least one request, DFSMSrmm displays the Exit Menu when you exit DFSMSrmm. From the Exit Menu, you can select options for processing the data set containing your saved commands. See “Exiting the Dialog” on page 50 for more information.

Changing List Sorting Options

You can change the sort order used by DFSMSrmm to build lists. You can change the fields to sort, and the direction (ascending or descending) they are sorted.

Use table field names to identify which fields to sort. These are the names of the different data columns, each of which contains specific information recorded by DFSMSrmm for a particular resource. For example, for each rack or bin number defined to DFSMSrmm, DFSMSrmm records a rack or bin number, unit ID, volume serial number, and rack or bin number status. Table field names differ between lists because DFSMSrmm records different information for different resources. Thus, DFSMSrmm sorts each type of list differently.

DFSMSrmm sets all sort options initially. After you change these settings, DFSMSrmm stores them and remembers them across sessions. Within the DFSMSrmm ISPF dialog, you can use either list sort options panels or the SORT command to change sort order.

Using Sort Options Panels

The dialog provides you with a sort options panel for each type of list you can build:

- Data set lists
- Software product lists
- Rack and bin number lists
- Volume lists
- Vital record specification list

Use the Dialog Sort Options Menu to select the list sort options panel you need. To display this menu, either:

- Select Option 2 (SORT) on the Dialog Options Menu and press ENTER.

or

- Type `OPTIONS SORT` on the command line of any DFSMSrmm panel and press `ENTER`.

Figure 21 shows the Dialog Sort Options Menu that DFSMSrmm displays:

```

Panel  Help
-----
EDGP@OP2          DFSMSrmm Dialog Sort Options Menu
Option ==>  1

1  DATA SET  - Specify Data Set list sort options
2  PRODUCT   - Specify Product list sort options
3  RACK      - Specify Rack list sort options
4  VOLUME    - Specify Volume list sort options
5  VRS       - Specify Vital Record Specifications list sort options

Enter selected option or END command.  For more info., enter HELP or PF1.


5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993

```

Figure 21. DFSMSrmm Dialog Sort Options Menu

Select the list sort option panel you need, based on the type of list you are sorting. For example, enter 1 to select the Data Set List Sort Options panel, as Figure 22 shows.

```

Panel  Help
-----
EDGP@STD          DFSMSrmm Data Set List Sort Options
Command ==>

    Enter the relative sort priority ( 1 to 5 ) and direction ( A or D ):

Table field name      Priority      Direction
-----
Data set name
Volume serial
Owner
File sequence
Create date

Enter END command to save changes, or CANCEL to end without saving.

```

Figure 22. DFSMSrmm Data Set List Sort Options Panel

Now specify the sort priorities and sort directions to set how all data set lists you request should be sorted. All of the list sort options panels are similar, except for the table field names used in the different DFSMSrmm lists. For more information on using the sort options panels and on the various table field names, see "How to Sort a List" on page 176.

Using the SORT Command

Use the SORT command from the command line of any list to change the sort order of the displayed information. Type SORT followed by one or more pairs of two parameters:

```
====> SORT <direction field> . . .
```

where

direction is Ascending or Descending, or an abbreviation (A or D).

field is a table field name, or unique abbreviation, of a data column on a specific list.

For example, typing

```
====> SORT ASCENDING UNIT
```

on the command line of a volume list causes DFSMSrmm to sort the items in the list by unit type in ascending order.

Table field names for data columns vary according to the kind of list you are sorting. See "How to Sort a List as You View It" on page 179 for more information on the table field names you can use with the SORT command to sort volume, data set, rack and bin number, software product and vital record specification lists.

Using Dialog Panels and Fast Path Commands

This section describes how to use menus and fast path commands to navigate through the dialog and to request functions. It also describes the help facility.

Navigating Through DFSMSrmm

DFSMSrmm provides an action bar-driven interface that exploits many of the usability features of Common User Access (CUA) interfaces. All screens are mixed case and most screens have action bars at the top. Some screens are longer than 24 lines and you can scroll through them. When screens are longer than 24 lines, issue the command PFSHOW OFF so that PF keys are not displayed on the screen.

Navigating through DFSMSrmm without Using the Action Bar

You can still navigate through DFSMSrmm using the standard method of typing in a selection number and pressing Enter.

Navigating through DFSMSrmm Using the Action Bar

Most DFSMSrmm panels have action bars at the top. The choices display in white (by default).

The action bar gives you another way to move through DFSMSrmm. If the cursor is located somewhere on the panel, there are several ways to move the cursor to the action bar:

- Using the keyboard's tab key
- Using mouse button
- Using the cursor manually

After you have chosen an ACTION, press ENTER to open the menu.

See "How to Use Line Operators" on page 182 for information on requesting functions using list line operators from lists, and "Chapter 12. Using RMM TSO Subcommands" on page 221 for information on requesting functions using the TSO command and subcommands.

Using Menus

The DFSMSrmm dialog offers several types of menus:

- User menus that are tailored to the tasks required by the various types of DFSMSrmm users
- A command menu that provides access to all functions
- Menus listing the functions you can request for each type of resource about which DFSMSrmm keeps information, such as shelf locations, volumes, data sets, owners, software products, and vital record specifications

User Menu

DFSMSrmm provides different user menus for general users, system programmers, storage administrators, and tape librarians. User menus are tailored specifically to the needs and access levels of each type of user. For more information on the user tasks you can perform from user menus, see "Who Can Use DFSMSrmm and How?" on page 20.

Your installation defines your level of access to functions. Using the menu designed for your user group is the best assurance that your requests for functions match the level of access you have been authorized to. In most cases, it is also the shortest path to the functions you want to request. If you request a function for which you are not authorized, your request will fail.

You can select a particular user menu from the DFSMSrmm primary option menu, or you can issue a fast path command to go directly to the menu. See "Issuing Fast Path Commands" on page 49 for more information on the specific fast path commands for user menus.

From the primary option menu, either:

- Select a menu number and press ENTER.
- or
- Issue a fast path command from the command or option line of any panel, and press ENTER.

Command Menu

All users can use the Command Menu to request any function menu, listing all the available functions for a particular resource. You can also request the Control Menu, from which you can display DFSMSrmm parmlib options and control information in the control data set.

To display the Command Menu, either:

- Select 5 on the DFSMSrmm primary option menu and press ENTER.
- or
- Type COMMANDS on the command or option line of any panel and press ENTER.

Figure 23 shows the Command Menu DFSMSrmm displays:

```
Panel Help
-----
EDGP@CMD                      DFSMSrmm Command Menu
Option ==>

0  OPTIONS - Specify dialog options and defaults
1  VOLUME  - Volume commands
2  RACK    - Rack and Bin Commands
3  DATA SET - Data Set commands
4  OWNER   - Owner commands
5  PRODUCT - Product commands
6  VRS     - Vital Record Specifications
7  CONTROL - Display system control information

Enter selected option or END command. For more info., enter HELP or PF1.

5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 23. DFSMSrmm Command Menu

Function Menus

DFSMSrmm records information about the following resources:

- Volumes
- Shelf locations
- Data sets
- Owners
- Software products
- Vital record specifications
- Parmlib options and control data set control information

Function menus list all the functions available for each type of resource about which DFSMSrmm records information. For example, the Volume Menu, shown in Figure 24 on page 49, lists all the functions you can request for volumes.


```
Panel  Help
-----
EDGPT000                                DFSMSrmm Volume Menu
Option ==>

0  OPTIONS - Specify dialog options and defaults
1  DISPLAY - Display volume information
2  ADD     - Add a new volume
3  CHANGE  - Change volume information
4  RELEASE - Delete or Release a volume
5  SEARCH  - Search for volumes
6  REQUEST - Request a volume
7  ADDSCR  - Add one or more SCRATCH volumes
8  CONFIRM - Confirm librarian or operator actions

Enter selected option or END command.  For more info., enter HELP or PF1.

5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 24. DFSMSrmm Volume Menu

You can access some function menus through user menus or through the Command Menu. Remember, however, that while you can access any function menu, you might not be authorized to request the available functions. If you are not authorized to request a function, your request fails.

Issuing Fast Path Commands

To help you navigate more quickly through the dialog, you can use fast path commands to display specific function panels. You issue these fast path commands from the command or option line of any panel, eliminating the need to scroll through panels or use menus to make your selections.

If you use a fast path command with no parameter, DFSMSrmm displays a menu from which you can make further selections. If you issue a command with a parameter, DFSMSrmm displays the function panel you requested. For example, if you type:

```
====> VOLUME
```

DFSMSrmm displays the Volume Menu from which you can request any volume function. (See Figure 24 for an example of this panel.)

If you type:

```
====> VOLUME DISPLAY
```

DFSMSrmm displays the Volume Display panel (Figure 25 on page 50), which you use to request information about a specific volume.

Panel	Help

EDGPT100	DFSMSrmm Volume Display
Command ==>	
Volume serial . . .	
Initial display . .	General information (Default) Access details Volume STATistics STOrage location details
Enter the volume serial and press ENTER to display volume information.	

Figure 25. DFSMSrmm Volume Display Panel

You can issue fast path commands consecutively to access new function panels or menus. DFSMSrmm nests each panel you request through a fast path command, retaining your place in the dialog and allowing you to return to a previous panel without losing any information you have entered. To return to a previous panel, press PF3 or use the END command to scroll back through the panels or menus you requested using fast path commands.

Use the fast path commands listed in “Appendix. DFSMSrmm ISPF Dialog Fast Path Commands” on page 433 to display specific panels.

Using Help Panels

Help panels are provided to explain specific tasks, describe fields on panels, and provide examples. Press PF1 or issue the HELP command from the command or option line of any function panel to view one or more help panels.

Exiting the Dialog

You can stop your DFSMSrmm session by:

- **Quick Exiting**

To completely exit the dialog, either:

Type =X on the command or option line of any panel and press ENTER.

or

Type X on the primary option menu and press ENTER.

You exit without specifying options for processing saved requests, and the data set containing any saved commands is available outside the dialog.

- **Exiting One Panel at a Time**

Use the END command or press the PF3 (END) function key from any panel to exit the dialog one panel at a time. If you used fast path commands to request several panels consecutively, you will return to those panels before exiting the dialog.

- **Exiting to Process Saved Requests**

Use the RETURN command or press PF4 to see the initial selection panel that appeared when you invoked DFSMSrmm. By default, this is the DFSMSrmm primary option menu. If you issued a fast path command before using RETURN or PF4, DFSMSrmm returns you to the panel from which you issued the fast path command.

Use the END command or press PF3 to exit the dialog from the initial selection panel and go to the Exit Menu, as Figure 26 shows.

The Exit Menu presents you with options for handling the data set in which DFSMSrmm saves your commands. You can use this menu if you issued the SAVE ON command or used background processing for your session on the Dialog User Options panel, and DFSMSrmm saved at least one command during your session.

```

Panel  Help
-----
EDGP@XIT                      DFSMSrmm Exit Menu
Command ==>>

    B - Browse Data Set          D - Delete Data Set
    E - Edit Data Set           K - Keep Data Set
    R - Rename Data Set         S - Submit batch job
    X - Execute in foreground

Commands have been saved in Data Set:

Data Set name . . . : 'RMML06.RMM0.LIST'

Specify new Data set name, if "R" selected:

New Data Set name . .

Job Statement Information:
==> /*
==> /*
==> /*
==> /*
Press ENTER to Process request, or END command to Exit,

```

Figure 26. DFSMSrmm Exit Menu

From the Exit Menu you can browse, edit, rename, delete or keep the data set containing your saved requests. You can also choose to process these requests in the foreground or you can submit a batch job.

To submit a batch job:

1. Enter the job statement information on the panel:
 - Include any job parameters necessary to your installation.
 - Enter /* as the first three characters for any unused job statement information lines.

2. Press ENTER to submit your job.

If you decide to defer submitting your job, use the PF3(END) key to leave the Exit Menu. DFSMSrmm saves the data set containing your requests with a name in the following format:

<tsa prefix>.RMMnn.LIST

where *nn* is a two-digit number of an ISPF logical screen. For example, a possible data set name for user WOOD could be WOOD.RMM10.LIST. Write the necessary JCL to create the job stream and submit the job outside the dialog.

Chapter 4. Defining Your Resources

This chapter describes how to use the DFSMSrmm ISPF dialog to define resource information to the DFSMSrmm control data set so that DFSMSrmm can manage those resources. You can define information about shelf locations, volumes, owners, software products, and data sets to DFSMSrmm. Once a volume is defined, DFSMSrmm automatically records information about it, as well as information about any data sets residing on it each time a data set is opened and closed.

Throughout this chapter, you are referred to help panels for additional information, and to the RMM TSO subcommands you can use instead of panels. See “Chapter 12. Using RMM TSO Subcommands” on page 221 for more information on using TSO subcommands.

Defining Shelf Locations

In your installation, you store your tape volumes and other removable media in shelf locations defined to DFSMSrmm by rack and bin numbers. Rack numbers define individual shelf locations in your removable media library. Rack numbers are six alphanumeric or national characters in any combination, such as rack number AB0001. Bin numbers define individual shelf locations in storage locations outside the removable media library. Your installation can use the DFSMSrmm built-in storage locations, LOCAL, DISTANT, or REMOTE or define storage locations with media name, management type, location name, and installation defined bin numbers. The bin numbers in the DFSMSrmm built-in storage locations are six numbers and are assigned by DFSMSrmm. Bin numbers in installation defined storage locations are six alphanumeric characters and are assigned by your installation.

Each time you add shelves in the removable media library or in a storage location, you must add individual rack and bin numbers to DFSMSrmm to define each new shelf location. When a new volume is stored in the removable media library, or when a volume is moved to a storage location, an empty shelf location must be available and already defined to DFSMSrmm by a rack or bin number.

Rack numbers sharing a common prefix can make up a *pool* in your removable media library. Pools are defined at installation time and can help you organize and manage the volumes in your shelves. For example, all rack numbers with a prefix of AB might make up a pool for temporary customer volumes, and all rack numbers with a prefix of SP might make up a pool for your software products.

To add rack numbers to a library in the removable media library, add them to a pool defined for that specific library.

Installation defined storage locations can be subdivided based on the media that resides in the location. For example, you can identify part of a storage location for cartridges and another part for reels. Provide a media name when you add bin numbers so the volumes are sent to the correct part of the storage location.

Adding New Rack and Bin Numbers

Related TSO Subcommand

Use the subcommands ADDRACK or ADDBIN to add rack or bin numbers to DFSMSrmm. See “ADDRACK: Adding a Shelf Location” on page 238 or “ADDBIN: Adding a Bin Number in a Storage Location” on page 224 for more information.

To add one or more rack numbers to the removable media library, or one or more bin numbers to a storage location:

1. Select Option 2 (ADD) on the Rack and Bin Menu and press ENTER.
DFSMSrmm displays the Add Racks and Bins panel, as Figure 27 shows:

Panel	Help

EDGPR200	DFSMSrmm Add Racks and Bins
Command ==>	
Specify one of:	
Rack number	Enter first or only rack number to be added
or	
Bin number	Enter first or only bin number to be added. Leave blank when adding built-in storage location bin numbers.
Location	Specify storage location name. Leave blank to add rack numbers.
Media Name	Required only for adding installation-defined storage location bin numbers.
Count of racks or bins . . .	Specify number of rack or bin numbers to add. Leave blank when adding only one rack or bin number.

Figure 27. DFSMSrmm Add Racks and Bins Panel

2. Enter the required information.
If you are adding rack numbers:
 - Enter a rack number if you are adding only one rack number to the removable media library.
 - Enter a rack number and the number of rack numbers to be added if you are adding several rack numbers.
If you are adding bin numbers:
 - Enter a location if you are adding one bin number to a built-in storage location.
 - Enter a location and the number of bin numbers to be added if you are adding more than one bin number to a built-in storage location.
 - Enter a location, initial bin number, and the number of bin numbers to be added if you are adding one or more bin numbers to an installation defined storage location.

If you are adding more than one rack or bin number, DFSMSrmm treats the number you supply as the initial rack number. Be sure that this number is long enough and low enough to accommodate the count value to be added to it without exceeding the numeric capabilities of the suffix.

You can use pool IDs to add one or more rack numbers to a specific library in the removable media library. For example, if pool KD* has been defined for the manual tape library dataserver, LIB1, you can use an initial rack number beginning with KD to add rack numbers to a pool in that specific library.

To add rack numbers to a pool, use the pool ID, or common prefix for rack numbers in that pool. Type CONTROL VLPOOLS from the command or option line of any panel to view information about the pools your installation has defined. To segregate shelf space in an installation defined storage location, use the media name defined in the location definition for the storage location. To obtain information about the installation defined storage locations, type CONTROL LOCDEF from the command or option line or use the LISTCONTROL LOCDEF subcommand.

3. Press ENTER to add the rack or bin numbers you specified to DFSMSrmm.

Deleting Rack and Bin Numbers

Related TSO Subcommand

Use the subcommands DELETERACK or DELETEBIN to delete rack or bin numbers from DFSMSrmm. See “DELETERACK: Deleting Shelf Location Information” on page 317 or “DELETEBIN: Deleting Bin Number Information” on page 308 for more information.

You can delete a rack number in the removable media library or a bin number in a storage location once it is empty and is no longer needed. A rack or bin number is empty when a volume is not currently stored at that shelf location. To delete one or more empty rack or bin numbers:

1. Select Option 4 (DELETE) on the Rack and Bin Menu and press ENTER.
DFSMSrmm displays the Delete Racks and Bins panel, as Figure 28 on page 56 shows:

Panel Help	

EDGPR400	DFSMSrmm Delete Racks and Bins
Command ==>	
Specify one of:	
Rack number	Enter first or only rack number to be deleted
or	
Bin number	Enter first or only bin number to be deleted. Leave blank when deleting built-in storage location bin numbers.
Location	Specify storage location name. Leave blank to delete rack numbers.
Media Name	Required only for deleting installation-defined storage location bin numbers.
Count of racks or bins . . .	Specify number of rack or bin numbers to delete. Leave blank when deleting only one rack or bin number.

Figure 28. DFSMSrmm Delete Racks and Bins Panel

2. Enter the required information:

If you are deleting rack numbers:

- Specify a rack number if you are deleting only one rack number from the library.
- Specify a rack number and how many rack numbers to be deleted if you are deleting several rack numbers.

If you are deleting bin numbers:

- Specify a location and a count if you are deleting bin numbers from a built-in storage location.
- Specify an initial bin number, a location, media name and how many bin numbers to be deleted if you are deleting several bin numbers from an installation defined storage location.

3. Press ENTER to delete any information you specified about the rack or bin numbers, such as the media name.

If you are deleting more than one bin number from an installation defined storage location, DFSMSrmm treats the number you specify as the initial bin number. When deleting bin numbers from the built-in storage locations, DFSMSrmm starts from the highest number and deletes bin numbers until it reaches either the count you specified or a shelf location containing a volume. DFSMSrmm stops when it reaches a shelf location containing a volume and issues an error message saying that it cannot delete all the bin numbers you requested.

Defining Volumes

When you manually define new volumes to DFSMSrmm, you must supply the volume serial number and volume status. You can supply more information or let DFSMSrmm automatically record information each time a data set on the volume is opened. DFSMSrmm automatically records:

- Information about how often a volume is used and resets this information when a volume is released and reused
- If permanent or temporary errors were encountered
- A security classification for the volume based on the highest security classification for the data sets residing on the volume
- Information about data sets on the volume if data set recording is on and if the data set is the first data set on the volume or if preceding data sets on the volume have been defined

You can use the DFSMSrmm sample exit EDGUX100 to control the data set information that DFSMSrmm records. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information.

DFSMSrmm can obtain information for volumes residing in a system-managed library from the tape configuration data base (TCDB) when you use the VOLCAT volume status when you define the volume to DFSMSrmm. DFSMSrmm also updates the TCDB each time the volume status changes. See “ADDVOLUME: Adding Volume Information” on page 241 for the volume information DFSMSrmm obtains from the TCDB.

You can manually define new non-scratch and scratch volumes to DFSMSrmm. Non-scratch volumes, also called private volumes, can have a status of *master* or *user*. For master volumes, DFSMSrmm allows the overwriting of data based on criteria set by the EDGRMMxx MASTEROVERWRITE operand. For user volumes, DFSMSrmm allows overwriting of data even if the data set name does not match the data set name of the existing file on the volume that is being overwritten. Scratch volumes are volumes that are available for use because there is no data on them, or the data on them has expired or is no longer needed.

When you manually define volumes, you can indicate where they should reside in your removable media library. Use either a rack number, representing a specific location on a shelf, or a pool ID to identify a volume’s shelf location. Your installation defines the pools to which you can associate volumes. If you specify neither a pool ID nor a rack number, or if you specify a pool ID and the prefix matches the volume serial number, DFSMSrmm attempts to use a rack number matching the volume serial number. The rack number or pool ID of the shelf location where the volume is to reside must be already defined to DFSMSrmm before you can define the volume.

Adding a New, Non-Scratch Volume

Related TSO Subcommand

Use the ADDVOLUME subcommand to add information about a single, new, non-scratch volume to DFSMSrmm. See “ADDVOLUME: Adding Volume Information” on page 241 for more information on ADDVOLUME.

To add a new, non-scratch volume:

1. Select Option 2 (ADD) on the Volume Menu and press ENTER.

Note: Tape librarians can bypass the Volume Menu by selecting option 6 (ADDVOL) on the Librarian Menu.

DFSMSrmm displays the DFSMSrmm Add Volume panel as shown in Figure 29.

Panel Help	

EDGPT210	DFSMSrmm Add Volume
Command ==>	
Volume N00001	Pool More: +
Count	or
Media name CART	Rack
Storage group	Location name . . .
Retention period . . 90	Expiration date . . . YYYY/DDD
Status USER	Initialize
Description	
Data set name	
Media type CST	Release actions
Label SL	Return to SCRATCH pool . . YES
Current version . . 3	Replace volume NO
Required version . . 4	Return to owner NO
Density	Initialize volume NO
Recording format . .	Erase volume NO
Compaction	Notify owner NO
Attributes	
Owner OLDJOE	Owner access
Assigned date	YYYY/DDD Assigned time . . .
Last read date	Last write date . . .
Security name	
Account number	
Jobname	MVS use YES
	VM use
Loan location	
Previous volume	
Volume access list:	Access
User User User User	
User User User User	
User User User User	
Enter SCROLL commands for more volume information, END to CANCEL.	

Figure 29. DFSMSrmm Add Volume Panel

2. Specify a volume serial number and a volume status. All other fields are optional. Press PF1 for field-specific help information.

Use either a pool ID or rack number to specify where the volume should reside in the removable media library. If you add a non-scratch volume with a volume serial number that is less than six characters, you must specify either a rack number or a pool ID.

To view the pool IDs defined by your installation, type CONTROL VLPOOLS from the command or option line of any panel. In the TSO environment, use the LISTCONTROL VLPOOL subcommand.

3. Scroll down to view additional fields and specify the volume information you want DFSMSrmm to record about the volume.
4. Press ENTER to add the volume to the DFSMSrmm control data set.

Adding Scratch Volumes

Related TSO Subcommand

Use the ADDVOLUME subcommand to add information about one or more scratch volumes to DFSMSrmm. See “ADDVOLUME: Adding Volume Information” on page 241 for more information on ADDVOLUME.

To define one or more scratch volumes to the removable media library:

1. Select Option 7 (ADDSCR) on the Volume Menu and press ENTER.

Note: Tape librarians can bypass the Volume Menu by selecting option 7 (SCRATCH) on the Librarian Menu.

DFSMSrmm displays the Add Scratch Volumes panel, as Figure 30 shows:

Panel	Help

EDGPT230	DFSMSrmm Add Scratch Volumes
Command ==>	
Volume	MAX001 Pool
	or
Media name	Rack
	Location name . . .
Count	(Default is 1)
Description	
Assigned Date . . .	(YYYY/DD) MVS Use
Assigned Time . . .	VM Use
Media type	
Label	(AL, NL or SL)
Density	(1600, 3480, 6250)
Initialize	(Default is YES)
Press ENTER to ADD one or more SCRATCH volumes, or END command to CANCEL.	

Figure 30. DFSMSrmm Add Scratch Volumes Panel

2. Supply a volume serial number.

You must specify COUNT if you are adding more than one scratch volume. DFSMSrmm treats the volume serial number you indicate as the initial volume serial number.

All other fields are optional.

Use either a pool ID or a rack number to direct where the scratch volumes should be stored in the removable media library. The rack number must be already defined to DFSMSrmm.

3. Press ENTER to add one or more scratch volumes to the DFSMSrmm control data set.

Redefining a Volume Already Defined to DFSMSrmm

If a volume with the same volume serial number as one you are adding is already defined to DFSMSrmm, you must either redefine the old volume before you add the new volume, or use a different serial number to add the new volume. To do this:

1. List information about the volume you are redefining, using the Display Volume panel or the LISTVOLUME subcommand. Knowing this information will help you correctly redefine the volume.
2. Delete all information about the volume, using the Delete Volume panel or the RMM DELETEVOLUME subcommand.
3. Manually add the volume. Specify a different volume serial number, preferably the rack number identifying the shelf location where the volume is currently stored, and a label type of NL. You can use the description field to identify the volume's real serial number and label type.

If your installation uses the installation exit EDGUX100, you do not have to redefine the volume serial number to use it on the system. For example, you can define a RACF profile, STGADMIN.EDG.IGNORE.TAPE.volser and authorize the user to this profile. The user should code EXPDT=98000 and VOLSER=rack_number in the JCL to use the volume. DFSMSrmm ignores the volume so it can be used. DFSMSrmm does not, however, track volume use and performs no management functions for the volume. For information on using the EDGUX100 installation exit to ignore known volumes or to manage volumes with duplicate volume serial numbers, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

Changing Volume Information

Related TSO Subcommand

Use the CHANGEVOLUME subcommand to change details for a volume already defined to DFSMSrmm. See "CHANGEVOLUME: Changing Volume Information" on page 282 for more information.

If DFSMSrmm recorded volume information when a data set on the volume was opened and closed, you are limited in the details you can change. See "Rules for Changing Volume Information" on page 282 for more information about the details you can change if you have CONTROL access to the STGADMIN.EDG.MASTER security resource.

You can also change volume information by using the RMM CHANGEVOLUME FORCE command. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.

To change information about a non-scratch volume already defined to DFSMSrmm:

1. Select Option 3 (CHANGE) on the Volume Menu and press ENTER.
DFSMSrmm displays the Change Volume panel as shown in Figure 31 on page 61:

Panel Help

EDGPT300 DFSMSrmm Change Volume

Command ==>

Volume serial . . . A00001

Enter the serial number of the volume to be changed and press ENTER.

Figure 31. DFSMSrmm Change Volume Panel

2. Enter the serial number of the volume for which you want to change information.
3. Press ENTER.

DFSMSrmm displays the Change Volume Details panel containing information about the volume you specified. For example, DFSMSrmm displays a panel like the one shown in Figure 32 on page 62⁶:

6. Import/Export support is available with APAR OW36342 or OW36343.

Panel Help

EDGPT310 DFSMSrmm Change Volume Details -

Command ==>

Volume : A00001

Media name : 3480

Storage group

Move mode : AUTO

Volume type : PHYSICAL

Retention period

Status : SCRATCH

Description

Data set name : 'SHELF.REPORT.DS001'

Media type *

Label : SL

Current version

Required version

Density *

Recording format

Compaction

Attributes

Availability

Owner

Assigned date : 1998/253

Last read date

Security name

Classification

Account number

Jobname

Loan location

Previous volume

Volume access list :

User

User

User

User

User

User

Pool

or

Rack : A00001

Location name

Destination

Bin number

In container

Expiration date . . . : 1999/015

Initialize : NO

Release actions:

Return to SCRATCH pool . .

Replace volume

Return to owner

Initialize volume

Erase volume

Notify owner

Expiry date ignore . . . : NO

Scratch immediate : NO

Owner access

Assigned time : 00:27:17

Last write date

MVS use : YES

VM use : NO

Last changed by : *HKP

Next volume

Access

User

User

User

User

User

User

More: +

Enter SCROLL commands for more volume information, END command to CANCEL.

Figure 32. DFSMSrmm Change Volume Details Panel

- Make changes or add missing information to any of the fields on the panel. You can scroll down to view additional fields and change volume information. Press PF1 for field-specific help information.
- Press ENTER to process your changes.

Deleting Volume Information

DFSMSrmm deletes volume information from the control data set when you release a volume. See “Releasing Volumes” on page 148 for more information.

Defining Owners

Owner IDs define users to DFSMSrmm. Owners can be individuals, departments, or any logical grouping of people. General users can change information for their own owner ID, once they have been defined to DFSMSrmm.

If a user who is not defined to DFSMSrmm runs a job that writes to a volume managed by DFSMSrmm, DFSMSrmm automatically records information about that user and adds an owner record for the user, using the user ID that requested the job as a DFSMSrmm owner ID. If you want DFSMSrmm to automatically notify the owner when a volume he owns is being considered for release, you must manually add the owner's user ID and node to this owner record.

When you delete an owner, DFSMSrmm prompts you to optionally provide a new user ID so existing volumes can be transferred to a new owner. The following sections describe how to add new owners, change owner information, and delete owners from DFSMSrmm.

Adding New Owners

Related TSO Subcommand

Use the ADDOWNER subcommand to define a new owner to DFSMSrmm. See "ADDOWNER: Adding Owner Information" on page 233 for more information on ADDOWNER.

Use the Owner Menu to add information about a new owner:

1. Select Option 2 (ADD) on the Owner Menu.
2. Enter an owner ID for the new owner and press ENTER.

We suggest that the owner ID coincide with an individual's RACF user ID or with the RACF group name of an application.

For example, if you specify an owner ID of OWN001, DFSMSrmm displays an DFSMSrmm Add Owner Details panel such as the one in Figure 33 on page 64.

Panel	Help

EDGP0200	DFSMSrmm Add Owner Details - OWN001
Command ==>	
Owner id	OWN001
Surname	NAKAMURA
Forenames	JOY
Department	M86
Address:	
Line 1	
Line 2	
Line 3	
Telephone:	
Internal	External
Electronic Mail:	
Userid	NAKAMURA
Node	NODE01
Press ENTER to ADD Owner Details, or END command to CANCEL.	

Figure 33. DFSMSrmm Add Owner Details Panel

3. Supply the information you want DFSMSrmm to record for this owner ID. Press PF1 for field-specific help information.
To ensure that the owner is automatically notified when an owned volume is released, you should enter both a user ID and a node, each of which can be up to eight characters. An owner's user ID might be different from his or her owner ID.

Changing Owner Information

Related TSO Subcommand

Use the CHANGEOWNER subcommand to change information for an owner ID previously defined to DFSMSrmm. See "CHANGEOWNER: Changing Owner Information" on page 277 for more information.

Note to General Users: You can only change information for your own owner ID.

To change any information about an owner ID previously defined to DFSMSrmm:

1. Select Option 3 (CHANGE) on the Owner Menu.
2. Type the owner ID for which you want to change information and press ENTER. DFSMSrmm displays a Change Owner Details panel for the owner ID you specified, such as the one shown in Figure 34 on page 65:

Panel	Help

EDGP0300	DFSMSrmm Change Owner Details - OWN000
Command ==>	
Surname	WOOD
Forenames . . .	JAMES
Department . . .	MS01
Address:	
Line 1	123 Rock Road
Line 2	San Jose
Line 3	CA
Telephone:	
Internal . . .	123-4567
External . . .	98765-5678
Electronic Mail:	
Userid	WOODJAME
Node	NODEB
Press ENTER to CHANGE Owner Details, or END command to CANCEL.	

Figure 34. DFSMSrmm Change Owner Details Panel

3. Change details or add missing information to any of the fields on the panel. Press PF1 for field-specific help information.
4. Press ENTER to process your changes.

Deleting Owner Information

Related TSO Subcommand

Use the DELETEOWNER subcommand to delete an owner ID from DFSMSrmm. See “DELETEOWNER: Deleting Owner Information” on page 313 for more information.

Note: Only tape librarians are authorized to delete an owner ID.

To delete an owner ID:

1. Select Option 4 (DELETE) on the Owner Menu.
2. Enter the owner ID that you want to delete, such as OWN000, and press ENTER.

If you did not request the confirm delete option and the owner ID you specified still owns volumes, DFSMSrmm displays a Delete Owner panel, such as the one in Figure 35 on page 66.

Panel Help

EDGP0420 DFSMSrmm Delete Owner - OWN000

Command ==>

New owner . . . WOODVICK

Transfer ownership of volumes by
entering the ID of the new owner.

Press ENTER to DELETE Owner, or END command to CANCEL.

Figure 35. DFSMSrmm Delete Owner Panel

3. If you want to optionally transfer ownership of existing volumes to another user, type the owner ID to which existing volumes should be transferred and press ENTER. The owner ID you type must already be defined to DFSMSrmm. DFSMSrmm then deletes the owner ID and transfers ownership of existing volumes to a new owner ID. If you don't want to transfer ownership to a new owner, delete the volumes using the DFSMSrmm ISPF dialog or RMM DELETEVOLUME subcommand.

Defining Software Products and Product Versions

This section describes how to define a new software product, or a new version of a software product, to the DFSMSrmm control data set. It also describes how to change and delete information about a software product.

Adding a New Software Product

Related TSO Subcommand

Use the ADDPRODUCT subcommand to add a new software product. See “ADDPRODUCT: Adding Software Product Information” on page 236 for more information on ADDPRODUCT.

To define a new software product or product version to DFSMSrmm:

1. Select Option 2 (ADD) on the Product Menu and press ENTER. DFSMSrmm displays the Add Product panel as shown in Figure 36 on page 67:

Panel	Help

EDGPP200	DFSMSrmm Add Product
Command ==>	
Enter Product Information:	
Product Number . .	
Level	Default V01R01M00
Press ENTER to ADD the Product, or END command to CANCEL.	

Figure 36. DFSMSrmm Add Product Panel

2. Enter a software product number.
All other fields are optional. Press PF1 for field-specific help information.
3. Press ENTER.
DFSMSrmm displays an Add Product Details panel showing the software product number you specified, such as the panel in Figure 37:

Panel	Help

EDGPP220	DFSMSrmm Add Product Details
Command ==>	
Enter Product information:	
Product Number . .:	PRODX01
Level	: V01R03M00 Owner . . . MAZOWN22
Product Name	
Description	
Add Volumes?	YES Yes or No
Press ENTER to ADD the Product details, or END command to CANCEL..	

Figure 37. DFSMSrmm Add Product Details Panel

4. Enter a software product name.
All other fields are optional. Press PF1 for field-specific help.
5. Press ENTER to add the new software product to DFSMSrmm.

Adding Software Product Volumes

If you specified YES in the Add Volumes? field on the Add Product Details panel, DFSMSrmm displays the Add Product Volume panel so you can add information about the volume.

```
Panel  Help
-----
EDGPP210          DFSMSrmm Add Product Volume          Product Added
Command ==>

Press ENTER to ADD the Product volume, or END command to CANCEL.
Product Number . .: PRODNUM

Level . . . . .: V01R03M00          Owner . .: MAZOWN22

Product Name . . .: PRODUCTION TIME

Description . . .:

Volume details:

      Volume  Rack/  Feature  Media  Location
      Serial  Pool   Code    Name   -----
      -----
VOL999  RACK99  9999    TAPE   SHELF
```

Figure 38. DFSMSrmm Add Product Volume Panel

1. As shown in Figure 38, enter a volume serial number, a rack number or pool ID, a feature code, a media name, and a location for each volume you want to associate with the software product. You can only enter information about one volume at a time. When you use the DFSMSrmm ISPF dialog to add software products, DFSMSrmm sets the retention period to 90 days, and the return to owner release action. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information on how to change these values if necessary.

Press PF1 for field-specific help information.

If a volume you specify has the same volume serial number as a volume already defined to DFSMSrmm, you might want to replace the old volume. For example, if you are adding a new release of a software product, you might want to consider disposing of the older release and deleting the volume information from DFSMSrmm. If you keep the older release, you must redefine the old software product volume before you can add the new software product volume, to avoid duplicate volume serial numbers.

See “Defining Volumes” on page 56 for more information on redefining volumes to avoid duplicate volume serial numbers.

2. Press ENTER to associate a volume with the product.

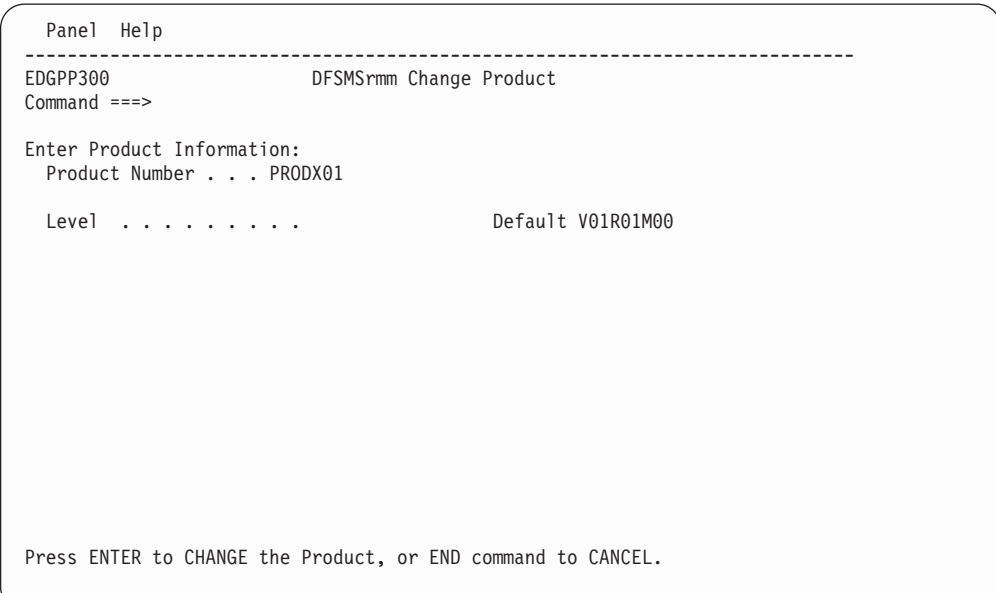
Changing Software Product Information

Related TSO Subcommand

Use the CHANGEPRODUCT subcommand to change information about a software product or version, or for volumes associated with the software product. See “CHANGEPRODUCT: Changing Software Product Information” on page 280 for more information.

To change information recorded by DFSMSrmm for a particular software product or version, or for associated volumes:

1. Select Option 3 (CHANGE) on the Product Menu and press ENTER.
DFSMSrmm displays the Change Product panel, as Figure 39 shows:



```
Panel  Help
-----
EDGPP300          DFSMSrmm Change Product
Command ==>

Enter Product Information:
Product Number . . . PRODX01

Level . . . . . Default V01R01M00

Press ENTER to CHANGE the Product, or END command to CANCEL.
```

Figure 39. DFSMSrmm Change Product Panel

2. Enter the software product number or identifier. Type in the version number if you are changing information for a specific version of the software product.
Press PF1 for field-specific help information.
3. Press ENTER.
DFSMSrmm displays a Change Product Details panel, such as the one in Figure 40 on page 70:

Panel Help

EDGPP310

DFSMSrmm Change Product Details

Row 1 to 1 of 1

Command ==>

Press ENTER to CHANGE Product details, or END command to CANCEL.

Product Number . .: PRODX01

Level: V01R03M00 Owner . . . MAZOWN22

Product NameTEST

DescriptionTEST

Volume details:

The following line commands are valid: A, D, R, V.

Volume Rack Feature

S Serial Number Code

VOL990 RAC990 4321

Scroll ==> PAGE

***** Bottom of data *****

Figure 40. DFSMSrmm Change Product Details Panel

- Change details or add missing information to any of the fields except:
 - Product Number
 - Level

If volumes have been identified for the software product, this panel includes volume details, such as volume serial numbers, rack numbers, and feature codes. For each volume listed, you can specify a line operator in the S column to request additional functions:

- A** Associate another volume with the software product.
- D** Delete information about an existing software product volume.
- R** Release a software product volume according to its release actions.
- V** Display complete information about a volume associated with a software product.

These line operators require different levels of authorization to be used effectively. Make sure you are authorized to request the function.

See “How to Use Line Operators” on page 182 for more information.

- Press ENTER to process your changes.

If you have added volumes associated with a product, use the RMM CHANGEVOLUME to change information about the volumes. For example, you might need to change the default retention period.

Deleting Software Product Information

Related TSO Subcommand

Use the DELETEPRODUCT subcommand to delete a product from DFSMSrmm and optionally release all volumes associated with it. See “DELETEPRODUCT: Deleting Software Product Information” on page 315 for more information.

To delete a product from DFSMSrmm and optionally release all volumes associated with it:

1. Select Option 4 (DELETE) on the Product Menu and press ENTER. DFSMSrmm displays the Delete Product panel, as Figure 41 shows.

Panel

Help

EDGPP400

DFSMSrmm Delete Product

Product Deleted

Command ===>

Enter Product Information:

Product Number . . . PRODY09

Level V01R03M00

Default V01R01M00

Release Product volumes Yes or No

Press ENTER to DELETE the Product, or END command to CANCEL.

Figure 41. DFSMSrmm Delete Product Panel

2. Enter the product number of the product you want to delete. Type in the version number of the product if you are deleting a specific version.
3. Specify YES in the Release Product volumes field if you want to release all volumes associated with the software product. If you specify NO, DFSMSrmm retains information about the volumes in the control data set, but no longer associates the volumes with the software product.
4. Press ENTER to delete the product and release all volumes, if requested.

If you specified the confirm delete option for your session, DFSMSrmm displays the DFSMSrmm Confirm Delete Product panel. This panel displays information about the product and allows you to confirm the deletion. If volumes have already been identified for the product, this panel includes volume details, such as volume serial numbers, rack numbers, and feature codes. DFSMSrmm displays a panel such as the one in Figure 42 on page 72:

Panel Help

EDGPP410

DFSMSrmm Confirm Delete Product

Row 1 to 2 of 2

Command ==>

Press ENTER to DELETE the Product, or END command to CANCEL.

Product Number . .: PRODX01

Level: V01R03M00 Owner . .: MAZOWN22

Product Name . . . : TEST

Description . . . : TEST

Volume details:

Volume Serial	Rack Number	Feature Code	
-----	-----	-----	
VOL490	RAC490	1234	
VOL990	RAC990	4321	

Scroll ==> PAGE

***** Bottom of data *****

Figure 42. DFSMSrmm Confirm Delete Product Panel

Press ENTER to delete the software product and release all volumes associated with it.

Defining Data Sets

Data sets can be defined to DFSMSrmm automatically when a data set on a volume is opened, or manually using the DFSMSrmm ISPF dialog or RMM ADDDATASET subcommand.

DFSMSrmm cannot record information about a data set if the first data set on the volume has not been defined to it. If a data set other than the first data set on the volume is being defined, any data sets preceding the data set you are adding must already be defined to DFSMSrmm.

When a data set is opened, DFSMSrmm records the:

- Data set name
- Record format
- Date and time the data set was created
- Date the data set was last accessed for read and write processing
- Data class, management class, storage class, and storage group for system-managed data sets

DFSMSrmm automatically records information about each data set on a volume. You can use the DFSMSrmm installation exit EDGUX100 so DFSMSrmm only records information about the first data set on the volume. Even though you are only requesting that DFSMSrmm records information about the first data set, you can still add information about other data sets on the volume using the RMM ADDDATASET subcommand.

Adding New Data Set Information

Related TSO Subcommand

Use the AD DD DATASET subcommand to add information about a data set to DFSMSrmm. See “AD DD DATASET: Adding Data Set Information” on page 227 for more information.

To manually add information about a new data set to DFSMSrmm:

1. Select Option 2 (ADD) on the Data Set Menu and press ENTER. DFSMSrmm displays the Add Data Set Details panel, as Figure 43 shows:

Panel	Help

EDGPD200	DFSMSrmm Add Data Set Details
Command ==>	
Data set name . . . MYDATA	
Volume serial . . . A00001	Physical file sequence number
Job name	Data set sequence number
Create date	YYYY/DDD
Create time	Record format
System id	Block Size
Date last read . . .	Logical record length . .
Date last written	Block count
	Unit address
Security name . . .	
Press ENTER to ADD the Data set details, or END command to CANCEL.	

Figure 43. DFSMSrmm Add Data Set Details Panel

2. Enter a data set name and a volume serial number. All other fields are optional. Press PF1 for field-specific help information.
3. Press ENTER to define the new data set to DFSMSrmm.

Changing Data Set Information

Related TSO Subcommand

Use the CHANGEDATASET subcommand to change data set information defined to DFSMSrmm. See “CHANGEDATASET: Changing Data Set Information” on page 270 for more information.

To change information recorded by DFSMSrmm for a particular data set:

1. Select Option 3 (CHANGE) on the Data Set Menu and press ENTER. DFSMSrmm displays the Change Data Set panel, as Figure 44 on page 74 shows.

Panel Help

EDGPD300 DFSMSrmm Change Data Set

Command ==>

Enter data set name and volume serial to be changed:

Data set name D027182.CHKEIL

Volume serial A00001

Physical file sequence number . . Default is 1

Figure 44. DFSMSrmm Change Data Set Panel

2. Enter a data set name and a volume serial number.
Specify the data set sequence number if the data set for which you want to change information is not the first data set on the volume.
Press PF1 for field-specific help.
3. Press ENTER.
DFSMSrmm displays a Change Data Set Details panel. This panel shows information defined to DFSMSrmm for the data set you specified. For example, DFSMSrmm displays a panel such as the one in Figure 45 on page 75:

Panel
Help

EDGPD310
DFSMSrmm Change Data Set Details

Command ==>

Data set name . . . : 'D027182.CHKEIL'
Volume serial . . . : A00001 Physical file sequence number . . : 2
Owner : D027182 Data set sequence number 0
More:

+

Job name
Step name
DD name
Create date 1998/351 YYYY/DDD
Create time 00:31:18
System id
Record format
Block size 0
Logical record length 0
Block count 0
Unit address

Date last read . . .
Date last written . .
Retention date . . :
VRS retained . . . : NO
VRS management value
Management class . . .
Data class
Storage class
Storage group

Security name VTAPE
Classification . . : VTAPE CLASS TO FORCE VOL ERASE

Primary VRS details:
VRS name . . . :
Job name . . . :
Subchain name :
VRS type
Subchain start date :

Secondary VRS details:
Value or class :
Job name . . . :
Subchain name :
Subchain start date :

Catalog status . . : UNKNOWN
Abend while open . . : NO

Press ENTER to CHANGE the data set name details, or END command to CANCEL.

Figure 45. DFSMSrmm Change Data Set Details Panel

4. Change details or add missing information to any of the fields on the panel.
Press PF1 for field-specific help.
If information was automatically recorded by DFSMSrmm when the data set was opened, you can change some information by using the RMM CHANGEDATASET FORCE command. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security profile and UPDATE access to STGADMIN.EDG.FORCE security profile.
5. Press ENTER to submit your changes.

Deleting Data Set Information

Related TSO Subcommand

Use the DELETEDATASET subcommand to delete data set information defined to DFSMSrmm. See “DELETEDATASET: Deleting Data Set Information” on page 311 for more information.

To delete information about a data set defined to DFSMSrmm:

1. Select Option 4 (DELETE) on the Data Set Menu. DFSMSrmm displays the Delete Data Set panel, as Figure 46 shows:

```
Panel  Help
-----
EDGPD400                      DFSMSrmm Delete Data Set
Command ===>

Enter data set name and volume serial to be deleted:

Data set name  . . . .
Volume serial  . . . .
Physical file sequence number . .          Default is 1
```

Figure 46. DFSMSrmm Delete Data Set Panel

Enter a data set name and a volume serial number.

Specify the data set sequence number if the data set is not the first data set on the volume. DFSMSrmm automatically deletes any information about data sets following the one you specified, but does not delete information about data sets preceding the one you specified. If you do not specify a data set sequence number, DFSMSrmm uses a default value of 1, so that information about all data sets on the volume is deleted.

Note: To delete a data set on a volume if DFSMSrmm recorded data set information during O/C/EOV processing, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security profile.

Press PF1 for field-specific help.

2. Press ENTER.

If you requested the confirm delete option for your session, DFSMSrmm displays a Confirm Delete Data Set panel. This panel displays information on the data set you

specified and asks you to confirm that you want to delete information about it. For example, DFSMSrmm displays a panel such as the one in Figure 47:

```

Panel  Help
-----
EDGPD410          DFSMSrmm Confirm Delete Data Set
Command ==>

Data set name . . . : 'D023427.DSNAME'
Volume serial . . . : VOL190      Physical file sequence number . . : 1
Owner . . . . . : MAZOWN22      Data set sequence number . . . : 1
Job name . . . . . : JOBNNAME
Step name . . . . . : STEP01      DD name . . . . . :
Create date . . . . : 1996/010    YYYY/DDD  Record format . . . . : FB
Create time . . . . : 08:05:37      Block size . . . . . : 3036
System id . . . . . : W98MVS2      Logical record length : 132
Date last read . . . : 1996/012     Block count . . . . . : 16
Date last written . . : 1996/012    Unit address . . . . : 0BE1
Retention date . . . : 1997/330     Management class . . .
VRS management value :              Data class . . . . . :
Matching VRS type . . :              Storage class . . . . :
Matching VRS job name :              Storage group . . . . :
Matching VRS name . . :
Security name . . . . : VTAPE
Classification . . . : VTAPE CLASS TO FORCE VOL ERASE
Press ENTER to DELETE the data set, or END command to CANCEL.
All details for following data sets on this volume will be deleted.

```

Figure 47. DFSMSrmm Confirm Delete Data Set Panel

Press ENTER to delete information about the data set you specified and all subsequent data sets on the volume. If your installation parmlib option is set to UNCATALOG(Y) in the DFSMSrmm parmlib member, DFSMSrmm also uncatalogs the data set, along with all subsequent data sets on the volume for which DFSMSrmm has recorded information in the DFSMSrmm control data set.

Chapter 5. Defining Vital Record Specifications

Use vital record specifications to set policies for retaining and moving all removable media. You can also use vital record specifications for moving data sets and volumes for disaster recovery and vital records within the removable media library and among your storage locations. The retention and movement policies specified in vital record specifications are processed during DFSMSrmm inventory management.

You can change vital record specifications to extend the retention period for a data set or volume or to change storage location information. You can also delete a vital record specification. If you change or delete a vital record specification, any currently scheduled moves must be completed before any new ones can take effect or the volume can be released.

Use the DFSMSrmm parmlib member EDGRMMxx OPTION operands to control the way DFSMSrmm performs vital record processing to apply movement and retention policies to data sets and volumes.

- Use the DFSMSrmm parmlib OPTION VRSCCHANGE operand to run trial run vital record processing when changes have been made to vital record specifications. As part of trial run processing, DFSMSrmm requires that you allocate an ACTIVITY file. The ACTIVITY file is not intended to be a report but contains detailed information about the changes made during vital record processing. Examine the ACTIVITY file to determine if the vital record specification changes you made are correct. DFSMSrmm provides a sample you can use to create the ACTIVITY report. The sample is described in *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.
- Use the DFSMSrmm parmlib OPTION VRSMIN operand to set the minimum number of vital record specifications that must be defined in order for you to run inventory management vital record processing. See “Defining the Minimum Required Policy” on page 136 for information about defining the minimum vital record specification
- Use the DFSMSrmm parmlib OPTION VRSEL operand to select the vital record processing functions you want performed. See “Which Type of Vital Record Processing Should You Use?” on page 130 for more information about selecting vital record processing.

See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information on DFSMSrmm inventory management and the DFSMSrmm parmlib OPTION command operands.

Types of Vital Record Specifications

There are three types of vital record specifications: data set, volume, and name. Data set and volume vital record specifications can be used separately, or in a chain with name vital record specifications.

Data Set Vital Record Specifications

A data set vital record specification defines the essential retention criteria by setting the total number of data set cycles to be retained, or the total number of days during which a data set should be retained. Data set vital record specifications are

defined using specific data set names, data set name masks, management class names, and vital record specification management values.

DFSMSrmm applies retention and movement policies to data sets by matching a vital record specification to a data set using the data set name in the vital record specification. It is possible for two vital record specifications to match to a data set in order to combine or merge retention and movement policy values. When this happens, the vital record specifications are called primary and secondary vital record specifications.

The primary vital record specification matches on data set name, data set name mask, management class name, or vital record specification management value.

The secondary vital record specification matches based on management class or vital record management value.

See “Identifying Data Sets for Retention” on page 81 for more information about defining data set vital record specifications.

Name Vital Record Specifications

Name vital record specifications can define additional move criteria and count for each location, but can only define retention criteria that does not override the maximum retention value set by the initial vital record specification. You can define name vital record specifications that can be chained to multiple data set or volume vital record specifications. Specify the DFSMSrmm parmlib `OPTION VRSEL(OLD)` operand if you want to define move information only; specify `VRSEL(NEW)` if you want to define either retention or movement information, or both.

Volume Vital Record Specifications

A volume vital record specification defines the essential retention criteria by setting the total number of volumes to be retained, or the total number of days during which a volume should be retained. Volume vital record specifications are defined using a specific or generic volume serial number. See “Identifying Volumes for Retention” on page 90 for more information about defining volume vital record specifications.

Chaining Vital Record Specifications

A vital record specification chain is a data set or volume vital record specification and all of the name vital record specifications chained from it. A vital record specification subchain starts with a data set vital record specification, name vital record specification with retention information, or a group of vital records chained using the RMM `ADDVRS ANDVRS` operand. The subchain includes all the vital record specifications chained from the start of the chain. The subchain ends before the next vital record in the chain that contains retention information. Both a vital record specification chain and a vital record specification subchain can be one or more vital record specifications.

Figure 48 on page 81 and Figure 49 on page 81 show a vital record specification chain where a data set vital record specification defines the policy to retain five cycles of data set A.*, one cycle to be stored in the home location. The `NEXTVRS`

operand in the vital record specification shown in Figure 48 points to N1.

```
RMM ADDVRS DSNAME('A.*') CYCLES COUNT(5) STORENUMBER(1) LOCATION(HOME) -  
NEXTVRS(N1)
```

Figure 48. Chaining Vital Record Specifications

This example shows the reuse of the name vital record specification N1 shown in Figure 49 to request that 4 cycles of the data set in the LOCAL storage location be retained.

```
RMM ADDVRS NAME(N1) STORENUMBER(4) LOCATION(LOCAL)
```

Figure 49. Specifying a Name Vital Record Specification for a Chain

The data set vital record specification defines five cycles as the maximum retention value. DFSMSrmm retains one cycle of data set A.* in the home location and four in the LOCAL storage location.

In Figure 50, a vital record specification is defined to retain a maximum of 20 cycles for the data set B.*. DFSMSrmm retains one cycle in the home location, ten cycles in the LOCAL storage location, and the last nine cycles back in the home location; extra cycles are retained in the home location by default.

```
RMM ADDVRS DSNAME('B.*') CYCLES COUNT(20) STORENUMBER(1) LOCATION(HOME) -  
NEXTVRS(N2)  
RMM ADDVRS NAME(N2) STORENUMBER(10) LOCATION(LOCAL)
```

Figure 50. Specifying a Vital Record Specification Chain

As part of DFSMSrmm inventory management vital records processing, DFSMSrmm verifies that the NEXTVRS specified in a vital record specification exists. If the NEXTVRS does not exist, DFSMSrmm issues an informational message EDG2221I and sets a return code of 4.

See “Chaining Retention and Movement Policies” on page 133 for information about chaining vital record specifications.

Identifying Data Sets for Retention

You can use fully qualified data set names, data set name masks, job names, and job name masks. You can manage generation data group (gdg) data sets and pseudo-gdg data sets using data set name masks.

During inventory management vital record processing, DFSMSrmm uses information in vital record specifications to identify the data sets that should be retained. DFSMSrmm compares the data set name information specified in vital record specifications against data set records in the control data set to apply policies. All the data sets with the same name that match a vital record specification are treated as a single vital record group.

You can optionally use a job name or job name mask in a vital record specification. DFSMSrmm then applies retention and movement policies based on a job name

that created a data set as well as a data set name. When job name is used, DFSMSrmm treats the set of data sets that match the vital record specification as a separate vital record group. See “How DFSMSrmm Applies Retention and Movement Policies” on page 83 for additional information.

Fully Qualified Data Set Names

You can define a retention policy for a specific data set by using a fully qualified data set name in a vital record specification. The vital record specification defined in this example defines the retention policy for one data set, PRITCHAR.BACKUP.DATA. All copies of the data set should be retained for five days.

```
RMM ADDVRS DSNAME('PRITCHAR.BACKUP.DATA') DAYS COUNT(5)
```

Data Set Name Masks

To manage multiple data sets with a single vital record specification, use data set name masks. DFSMSrmm movement and retention policies apply to any data sets matching the data set name mask you define. To create a data set name mask, use *, %, or ~ as placeholders.

* (asterisk)

A single * represents a single qualifier of any number of characters.

```
RMM ADDVRS DSNAME('PRITCHAR.BACKUP.*')
```

A single * when used within a qualifier represents zero or more characters.

```
RMM ADDVRS DSNAME('PRITCHAR.BACK*.DATA')
```

More than one single * can be used within a qualifier as long as a character precedes or follows the *.

```
RMM ADDVRS DSNAME('PRITCHAR.*A*B.DATA')
```

.* represents zero or more qualifiers. At the end of the mask, it indicates that any remaining characters are to be ignored.

```
RMM ADDVRS DSNAME('PRITCHAR.**')  
RMM ADDVRS DSNAME('PRITCHAR.**.DATA')
```

** indicates to select all data sets.

```
RMM ADDVRS DSNAME('**')
```

You can use this mask in a vital record specification to define installation default retention criteria for data sets that are not covered by other vital record specifications.

% (percent sign)

A place holder for a single character. You can use one or more % signs to create a data set name mask.

```
RMM ADDVRS DSNAME('PRITCHAR.BACKUP.DAT%')
```

~ (not sign)

A place holder for a single character in a data set name mask for a pseudo-GDG data set name. A pseudo-GDG is a collection of data sets, using the same data set name pattern. The ~ indicates that DFSMSrmm should manage data sets matching the data set name mask like a generation data group.

The following example defines a policy to retain the five most current versions of data sets matching the data set name mask.

```
RMM ADDVRS DSNAME('PRITCHAR.BACK--.DA--') CYCLES  
COUNT(5)
```

Use % rather than - as a place holder, when you do not want to manage all the data sets that match the data set name mask as a pseudo-GDG. See “Pseudo-GDG Data Set Names” on page 88 for more information about identifying pseudo-GDGs for retention and movement.

Management Class and Management Value Names

You can specify a management class name or vital record specification management value in a vital record specification. A management class name or management value can be one to eight alphanumeric characters, beginning with an alphabetic character, and must follow standard MVS data set naming conventions. This name must be a single qualifier. For example, you can define the management class name M99000 for all data sets with the special date 99000.

Special Data Set Name Masks

You can use the reserved data set names ABEND and OPEN to specify policies for data sets closed as a result of an abnormal end in a task or for data sets in use during inventory management.

```
RMM ADDVRS DSNAME('ABEND') LASTREFERENCEDAYS COUNT(5)  
RMM ADDVRS DSNAME('OPEN') JOBNAME(LAURAN) LASTREFERENCEDAYS COUNT(5)
```

Name Filtering with Data Set Name Masks

Use a data set name mask WOOD.** to define a retention and movement policy for all data sets that belong to user ID, WOOD. To retain all data sets belonging to the user, WOOD, for 99 days, specify:

```
RMM ADDVRS DSNAME('WOOD.**') DAYS COUNT(99)
```

You can then use a more specific data set name mask that matches a subset of all the data sets to retain the WOOD.BACKUP data sets. In this example, you are defining a vital record specification to retain data sets that match the data set name mask, WOOD.BACKUP.**, for 10 days.

```
RMM ADDVRS DSNAME('WOOD.BACKUP.**') DAYS COUNT(10)
```

How DFSMSrmm Applies Retention and Movement Policies

DFSMSrmm applies retention and movement policies to data sets based on the most specific match between the data set and job name masks in vital record specifications and the data set names and job names defined to DFSMSrmm. During vital record processing as described in *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*, DFSMSrmm matches data sets to vital record specifications. If there are no matches, no policies are applied to data sets. DFSMSrmm does not issue any messages if there are no matches.

Your installation can control how DFSMSrmm matches policies to data sets using the DFSMSrmm parmlib OPTION VRSEL operand and the OPTION VRSJOBNAME operand.

How DFSMSrmm Applies Policies When Using VRSEL(OLD)

DFSMSrmm matches policies to data sets based on the value you set for the DFSMSrmm EDGRMMxx parmlib member OPTION VRSEL operand.

Table 5 shows the order that DFSMSrmm uses to match data sets defined to DFSMSrmm to data set name masks in vital record specifications when VRSEL(OLD) is specified. It also shows the comparative order when VRSEL(NEW) is specified.

Table 5. How DFSMSrmm Matches Data Set Names to Data Set Masks

Order (OLD)		Description	Example
1	1	A reserved name, ABEND or OPEN matches a data set before any other data set mask.	RMM ADDVRS DSNAME('ABEND') LASTREFERENCEDAYS COUNT(1) RMM ADDVRS DSNAME('OPEN') LASTREFERENCEDAYS COUNT(2)
2	7	A management class name matches the management class value after ABEND or OPEN and before any other data set mask. The management class value is set using the RMM EDGUX100 installation exit and matches management class names set using ACS routines.	RMM ADDVRS DSNAME('M99000') WHILECATALOG
3	2	A fully qualified data set name matches a data set name before any mask containing %, ~, or *.	RMM ADDVRS DSNAME('PRITCHAR.BACKUP.DATA')
4	3	Any data set name mask that includes a % or a ~ matches a data set name before masks containing an *. % and ~ are treated equally.	RMM ADDVRS DSNAME('PRITCHAR.%.DATA') RMM ADDVRS DSNAME('PRITCHAR.~.DATA')
5	4	Any data set name mask that includes single *'s preceded or followed by any character is the next best match.	RMM ADDVRS DSNAME('PRITCHAR.*BACK.DATA')
6	5	Any data set name mask that includes an * is the next best match.	RMM ADDVRS DSNAME('PRITCHAR.*.DATA')
7	6	A data set name mask that includes ** anywhere in the mask is the next best match. When matching the data set mask to the data set name, the mask where the non-generic characters occur earlier in the mask is most specific.	RMM ADDVRS DSNAME('PRITCHAR.**') RMM ADDVRS DSNAME('**.PRITCHAR') RMM ADDVRS DSNAME('*.**')
8	8	A vital record specification management value set by using the DFSMSrmm EDGUX100 exit.	RMM ADDVRS DSNAME('D99000') WHILECATALOG
9	9	A data set name mask of ** indicates the policy applies to all data sets not managed by any other vital record specification.	RMM ADDVRS DSNAME('**')

DFSMSrmm uses a data set name mask to apply a policy before using a vital record specification management value except when the data set mask '**' is specified. If a vital record specification specified with a data set name mask and a vital record specification specified with a vital record specification management

value match to a data set, the vital record specification with the data set name mask is used unless the data set name mask is '***'. DFSMSrmm matches a data set to the most specific data set mask when multiple matches occur.

How DFSMSrmm Applies Policies When Using VRSEL(NEW)

Table 5 on page 84 shows the order that DFSMSrmm uses to match data sets defined to DFSMSrmm to data set name masks in vital record specifications when VRSEL(NEW) is specified. If DFSMSrmm matches a vital record specification based on data set name, DFSMSrmm then tries to match a management class to a data set and then a management value to a data set.

DFSMSrmm uses a data set name mask to apply a policy before using a vital record specification management value except when the data set mask '***' is specified. If a vital record specification specified with a data set name mask and a vital record specification specified with a vital record specification management value match to a data set, the vital record specification with the data set name mask is used unless the data set name mask is '***'. DFSMSrmm matches a data set to the most specific data set mask when multiple matches occur.

Job Names

You can use the name of the job that created the data set to retain a data set. A job name consists of one to eight alphanumeric characters, or \$, # or @. If your installation uses ISO/ANSI standard label volumes, job name can only contain alphanumeric characters and must not contain national characters.

For example, to keep all data sets created by the job LAURAN, the following vital record specification could be defined.

```
RMM ADDVRS DSNAME('***') JOBNAME(LAURAN)
```

Job Name Masks

You can use a job name mask to retain data sets based on the job that created the data set. Specify a job name using one to eight alphanumeric characters, or \$, # or @. To create a job name mask, use an * to match any character string and % to match any one character.

When matching data set and job names to vital record specifications, a job name is a more specific match than a job name mask. If you use a job name mask, DFSMSrmm identifies data sets that are created by jobs that match the job name mask and separates them into vital record groups. All unique job names matching a job name mask, are managed as a vital record group.

To create job name masks, you can use: * and %.

* (asterisk)

A single * matches all job names.

```
RMM ADDVRS DSNAME('***') JOBNAME(*)
```

DFSMSrmm treats * like any other job name mask. Data sets that are created by jobs matching * are separated into vital record groups.

A single * within a job name matches zero or more characters. You cannot specify ** in a job name mask.

```
RMM ADDVRS DSNAME('***') JOBNAME(S*)
```

You can use more than one * in a job name mask, as long as a character precedes or follows the *. This example matches all job names that start with 'S' and have a second 'S' somewhere in the name. For example, SS, SMSTAPE, and S123S are all valid job name matches.

```
RMM ADDVRS DSNAME('**') JOBNAME(S*S*)
```

% (percent sign)

A place holder for a single character.

```
RMM ADDVRS DSNAME('**') JOBNAME(S%)
```

The example matches all job names that start with 'S' and are exactly two characters. Job names that might match the mask include: S1, SX, and ST.

```
RMM ADDVRS DSNAME('**') JOBNAME(S%S*)
```

The example matches job names of three or more characters that have 'S' as the first and third characters. Job names that might match the mask include: S8S and SSS123.

Figure 51 shows examples of RMM ADDVRS subcommands using valid job name masks.

```
RMM ADDVRS DSNAME('MAXWELL.DATA.*') JOBNAME(STSG*)
RMM ADDVRS DSNAME('MAXWELL.DATA.*') JOBNAME(A%%XX)
RMM ADDVRS DSNAME('MAXWELL.DATA.*') JOBNAME(W*)
RMM ADDVRS DSNAME('MAXWELL.DATA.*') JOBNAME(XY%T*)
```

Figure 51. Valid Job Name Masks

How DFSMSrmm Uses Jobname to Apply Policies

Table 6 shows how DFSMSrmm applies policies based on the jobname you specify and the DFSMSrmm parmlib EDGRMMxx member OPTION VRSJOBNAME operand value you use.

Table 6. How DFSMSrmm Uses Jobname to Apply Policies

If You Specify	DFSMSrmm Uses	And if There Is No Match
VRSJOBNAME(1)	The data set and job name to match a data set to a vital record specification. Job name is the primary value used to match the data set to a vital record specification.	A match by data set name only is acceptable.
VRSJOBNAME(2)	The data set name to match a data set to a vital record specification. If a data set matches multiple vital record specifications with the same data set name, then DFSMSrmm uses a job name to further qualify the data set name.	DFSMSrmm does not apply a policy to the data set.

Table 7 on page 87 shows the order that DFSMSrmm uses to match job names defined to DFSMSrmm to job name masks in vital record specifications.

Table 7. How DFSMSrmm Matches Job Names to Job Name Masks

Order	Description	Example
1	A specific job name matches a job name before any job name mask containing special characters % or *.	RMM ADDVRS DSNAME('PRITCHAR.BACKUP.DATA') - JOBNAME(RMMJOB1)
2	Any job name mask that includes a % or an * matches a job name.	RMM ADDVRS DSNAME('PRITCHAR.BACKUP.BACK.DATA') - JOBNAME(RMMJOB*)
3	A job name of * matches to all data sets that have a job name specified.	RMM ADDVRS DSNAME('PRITCHAR.BACKUP.BACK.DATA') - JOBNAME(*)

Name Filtering with Job Name Masks

When you specify a job name or job name mask in a vital record specification, there can be a more specific match between vital record specifications and data sets.

```
RMM ADDVRS DSNAME('A.B*') CYCLES COUNT(3)
RMM ADDVRS DSNAME('A.B*') CYCLES COUNT(3) JOBNAME(J*)
```

In the first example, DFSMSrmm uses data set name only to match data sets to data set name masks. When there are multiple data sets of the same name that match the data set name mask, the sets of matching data sets are managed as a vital record group. There can be multiple vital record groups. For example, three A.B1 data sets are retained as one vital record group; three A.BX data sets are retained as another vital record group.

When you add a job name mask as shown in the second vital record specification, there can be more vital record groups because each set of data sets with the same data set name and job name is managed as a separate vital record group. For example, you could have separate vital record groups for data sets A.B1 created by JOB1; data sets A.B1 created by job JOB2; data sets A.B1 created by JOB3, and so on.

When you specify JOBNAME(1), DFSMSrmm concatenates the jobname mask and the data set name to form a single filter mask. When you specify JOBNAME(2), the data set name mask and jobname mask are concatenated to form a single filter mask. If a data set matches to multiple vital record specifications, DFSMSrmm selects the vital record specification that matches most specifically. The most specific match is determined by scanning the filter masks from left to right checking for generic characters (% , * , or **). The sooner a generic character is found in a filter mask, the less specific the match is.

For example, a data set MAXWELL.DATA.SET is created by a job with JOBNAME BANK. Two vital record specifications are defined:

```
RMM ADDVRS DSN('MAXWELL.DATA.SET') JOBNAME(B*)
RMM ADDVRS DSN('*.DATA.SET') JOBNAME(BANK)
```

The data set matches to both vital record specifications. With JOBNAME(1), the most specific match is determined by comparing the data set name as against the concatenated jobname and data set names,

```
B*.MAXWELL.DATA.SET
BANK.*.DATA.SET
```

BANK.*.DATA.SET is the most specific match.

With JOBNAME(2), the most specific match is determined by comparing the data set name as against the concatenated jobname and data set names,

```
MAXWELL.DATA.SET.B*  
*.DATA.SET.BANK
```

MAXWELL.DATA.SET.B* is the most specific match.

Generation Data Group (GDG) Base Names

You can use a GDG base name when defining vital record specifications to retain volumes. You must not supply the generation data set group suffix. You must specify CYCLES if you want DFSMSrmm to manage the data sets as a group. Figure 52 defines a retention policy for a GDG base name, SOTIRI.RETAIN.

```
RMM ADDVRS DSNAME('SOTIRI.RETAIN') GDG CYCLES COUNT(3)
```

Figure 52. Defining a Retention Policy for a GDG Base Name

The three most current generations are retained. For example:

```
SOTIRI.RETAIN.G0001V00  
SOTIRI.RETAIN.G0002V00  
SOTIRI.RETAIN.G0003V00
```

If you are using GDG version numbering, DFSMSrmm only keeps the latest version of each generation.

Pseudo-GDG Data Set Names

A pseudo-GDG is a collection of data sets, using the same data set name pattern, that DFSMSrmm manages like a GDG. A *pseudo-GDG* data set name contains the ~ as a placeholder for the characters in the pattern that change with each generation. You can use * or % in the pseudo-gdg data set name mask as placeholders, but ~ is the character that indicates the data set name mask is for a pseudo-gdg data set.

Figure 53 shows examples of RMM ADDVRS commands using data set name masks for pseudo-GDGs.

```
RMM ADDVRS DSNAME('RPR%.*.*.G~V~')  
RMM ADDVRS DSNAME('RPR%.*.V~.~.*')  
RMM ADDVRS DSNAME('PREF.A*.~')  
RMM ADDVRS DSNAME('PREF.*~CD.XY')  
RMM ADDVRS DSNAME('PREF.~*~.AB.**')  
RMM ADDVRS DSNAME('PREF.**.DATA~.*')
```

Figure 53. Specifying Data Set Name Masks for Pseudo-GDGs

The restrictions for using masking characters are:

- You cannot include a ~ character between two * characters in each data set qualifier. For example, you cannot use:

```
RMM ADDVRS DSNAME('JAMES.*DIV~*.SOCCER.**')
```


- You cannot include a ~ character between a pair of ** characters in each data set mask. For example, you cannot use:

```
RMM ADDVRS DSNAME('JAMES.**.DIV~.SOCCER.**')
```

During DFSMSrmm inventory management, DFSMSrmm recognizes the ~ masking character, and applies the appropriate retention and movement policy defined for a pseudo-GDG. DFSMSrmm validates the mask you use and rejects a vital record specification if the data set name mask is used incorrectly.

Some systems, like DB2 and DFSMSHsm, maintain their own generations by assigning a unique data set name to each member of the cycle or generation. With pseudo-GDGs, you can define a set of data sets to be managed like a GDG. In this example, the vital record specification defines a policy for a user-defined set of data sets which are sequenced like a GDG.

```
RMM ADDVRS DSNAME('PRITCHAR.X~Y~') -  
NOGDG CYCLES COUNT(3)
```

Note: You must use the NOGDG operand when defining a pseudo-GDG vital record specification.

Specifying GDG and Non-GDG Data Set Names

When defining policies to manage a GDG base entry and a standard data set name, you cannot use the same data set name in two vital record specifications. You also cannot have two vital record specifications that use the same data set name and job name combination when managing GDGs and non-GDGs.

If you want to define different retention and movement criteria for a data set name, you can define vital record specifications using different data set name masks. For one of the vital record specifications, specify a data set name mask using the GDG suffix together with ~. The other vital record specification can include a specific or a generic data set name. DFSMSrmm manages both sets of non-GDG data sets separately. When you include the GDG suffix and ~ in the data set name for the first vital record specification, DFSMSrmm manages those data sets as a cycle of pseudo-GDG data sets, even though they are identified as NOGDG.

Figure 54 shows an example of defining two vital record specifications for the same data set name with the CYCLES operand. When you use the CYCLES operands, duplicate GDG's are not retained by a vital record specification.

```
RMM ADDVRS DSNAME(a.b.c) NOGDG CYCLES COUNT(5) LOCATION(REMOTE)  
RMM ADDVRS DSNAME(a.b.c.G~V~) -  
NOGDG CYCLES COUNT(5)
```

Figure 54. Specifying NOGDG and CYCLES in a Vital Record Specification

Figure 55 on page 90 shows an example of defining two vital record specifications for the same data set name using the NOGDG and DAYS operands. When DAYS is specified the duplicate GDG's are retained.

```
RMM ADDVRS DSNAME(a.b.c) NOGDG DAYS COUNT(5) LOCATION(REMOTE)
RMM ADDVRS DSNAME(a.b.c.G- - - - -) -
      NOGDG DAYS COUNT(5)
```

Figure 55. Specifying NOGDG and DAYS in a Vital Record Specification

You must select a vital record specification type of DSNAME, VOLUME, or NAME.

Identifying Volumes for Retention

You can define retention policies for volumes by using a specific or generic volume serial number.

Retaining Volumes by Specific Volume Serial Number

Use a full volume serial number to retain a volume. A full volume serial number has one to six alphanumeric characters, or \$, #, and @, or special characters.

To retain the volume AS0001 for five days, issue:

```
RMM ADDVRS VOLUME(AS0001) COUNT(5)
```

Retaining Volumes by Generic Volume Serial Number

Use a generic volume serial number to retain multiple volumes. A generic volume serial has one to five characters followed by an asterisk. For example, if you define a vital record specification for volume AS00*, DFSMSrmm retains all volumes with serial numbers beginning with AS00.

To retain twenty volumes that match the AS00* generic volume serial number, issue:

```
RMM ADDVRS VOLUME(AS00*) COUNT(20)
```

Adding Vital Record Specifications

Related TSO Subcommand

Use the ADDVRS subcommand to add a vital record specification for a data set or volume, or to add a name vital record specification. See “ADDVRS: Adding a Vital Record Specification” on page 255 for more information.

To add a vital record specification:

1. Select Option 2 (ADD) on the Vital Record Specification Menu and press ENTER.

DFSMSrmm displays the Add Vital Record Specification panel, as Figure 56 on page 91 shows:

```
Panel  Help
-----
EDGPV200      DFSMSrmm Add Vital Record Specification
Command ==>

Specify one of the following:
  Data set mask . .
    Job name mask . .

  Volume serial . .      ( May be generic )

  VRS name . . . . VRSNAME
```

Figure 56. DFSMSrmm Add Vital Record Specification Panel

2. Enter a data set name and optionally a job name to add a data set vital record specification. Enter a volume serial number to add a volume vital record specification or a vital record specification name to add a name vital record specification. Press PF1 for help or for more specific help information.
3. Press ENTER.

DFSMSrmm displays the Add Data Set VRS panel, the Add Volume VRS panel, or the Add Name VRS panel.

Adding Data Set Vital Record Specifications

If you used a data set mask on the Add Vital Record Specification panel, DFSMSrmm checks to see if a vital record specification already exists for the data set, then displays the Add Data Set VRS panel as shown in Figure 57:

Panel Help	

EDGPV210	DFSMSrmm Add Data Set VRS
Data set mask : '**'	GDG . . NO
Job name mask :	
Count 5	Retention type CYCLES
	While cataloged yes
Delay 0 Days	Until expired no
Location OFFSITE	
Number in location . 5	
Priority 0	
	Release options:
Next VRS in chain . . NXT1	Expiry date ignore yes
Chain using . . . AND	Scratch immediate yes
Owner WOODMW	
Description . . .	
Delete date . . . 1999/365	(YYYY/DDD)
Press ENTER to ADD the VRS, or END command to CANCEL.	
Command ==>	

Figure 57. DFSMSrmm Add Data Set VRS Panel

Enter any information you want DFSMSrmm to record for the vital record specification. All fields on this panel are optional. Press PF1 for help.

Press ENTER to add the vital record specification.

Adding Volume Vital Record Specifications

If you entered a generic volume serial number on the Add Vital Record Specification panel, DFSMSrmm checks to see if a vital record specification already exists for the volume. It then displays the Add Volume VRS panel as shown in Figure 58:

Panel

Help

EDGPV230DFSMSrmm Add Volume VRS

Volume Serial : WOOD*

Count 99999

Delay 0 Days

Location OFFSITE

Number in location . 99999

Priority 0

Next VRS in chain . .

Owner WOODMW

Description . . . Keep all master volumes off-site

Delete date . . . 1999/365 (YYYY/DDD)

Press ENTER to ADD the VRS, or END command to CANCEL.

Command ==>

Figure 58. DFSMSrmm Add Volume VRS Panel

Enter any information you want DFSMSrmm to record for the vital record specification. All fields on this panel are optional. Press PF1 for help.

Press ENTER to add the vital record specification.

Adding Name Vital Record Specifications

If you entered a name on the Add Vital Record Specification panel, DFSMSrmm checks to see if a vital record specification with the same name is already defined. It then displays the Add Name VRS panel as shown in Figure 59:

```
Panel  Help
-----
EDGPV240          DFSMSrmm Add Name VRS

Name . . . . . : NXT1

Count . . . . . 2          Retention type . . . . . CYCLES
                          While cataloged . . . . . no
                          Until expired . . . . . no

Location . . . . . HOME
Number in location . 2

Next VRS in chain . . NXT2
Chain using . . . NEXT

Owner . . . . . WOODMW
Description . . .
Delete date . . . 1999/365      ( YYYY/DDD )

Press ENTER to ADD the VRS, or END command to CANCEL.
Command ==>
```

Figure 59. DFSMSrmm Add Name VRS Panel

Enter any information you want DFSMSrmm to record for the vital record specification. All fields on this panel are optional. Press PF1 for help.

Press ENTER to add the vital record specification.

Changing Vital Record Specifications

Related TSO Subcommand

To change information about a vital record specification using TSO subcommands:

1. Use the LISTVRS subcommand to view details about the vital record specification for which you want to change information.
2. Use the DELETEVRS subcommand to delete the vital record specification.
3. Use the ADDVRS subcommand to redefine the vital record specification.

See “LISTVRS: Displaying Information about a Vital Record Specification” on page 355, “DELETEVRS: Deleting Vital Record Specifications” on page 324, and “ADDVRS: Adding a Vital Record Specification” on page 255 for more information.

To change information defined to DFSMSrmm for a vital record specification:

1. Select Option 3 (CHANGE) on the Vital Record Specification Menu and press ENTER. DFSMSrmm displays the Change Vital Record Specification panel, as Figure 60 shows:

```
Panel  Help
-----
EDGPV300      DFSMSrmm Change Vital Record Specification
Command ==>

Specify one of the following:
  Data set mask . .
    Job name mask . .

  Volume serial . .      ( May be generic )

  VRS name . . . . VRSNAME
```

Figure 60. DFSMSrmm Change Vital Record Specification Panel

2. Enter a data set name, a volume serial number, or a vital record specification name.

Note: You must specify a data set name before you can specify a job name.
Press PF1 for help.

3. Press ENTER.

DFSMSrmm displays either the Change Data Set VRS panel, the Change Volume VRS panel, or the Change Name VRS panel.

If you entered a data set name on the Change Vital Record Specification panel, DFSMSrmm displays a Change Data Set VRS panel such as the one in Figure 61:

Panel

Help

EDGPV310

DFSMSrmm Change Data Set VRS

Data set mask : '**'

Job name mask :

GDG . . NO

Count 5

Retention type CYCLES

While cataloged yes

Delay 0 Days

Until expired no

Location OFFSITE

Number in location . 5

Priority 0

Next VRS in chain . . NXT1

Chain using . . . AND

Release options:

Expiry date ignore yes

Scratch immediate yes

Owner WOODMW

Description . . .

Delete date . . . 1999/365 (YYYY/DDD)

Press ENTER to CHANGE the VRS, or END command to CANCEL.

Command ==>

Figure 61. DFSMSrmm Change Data Set VRS Panel

Make changes or add missing information to any of the fields on the panel. Press PF1 for help. Press ENTER to process your changes.

Deleting a Vital Record Specification

Related TSO Subcommand

Use the DELETEVRS subcommand to delete a vital record specification defined to DFSMSrmm. See “DELETEVRS: Deleting Vital Record Specifications” on page 324 for more information.

When a vital record specification is deleted, no data set or volume information is changed. During the next vital records processing run, DFSMSrmm uses only the remaining vital record specifications to apply policies. If the data set or volume matches to another remaining vital record specification, DFSMSrmm applies those policies. If the data set or volume does not match to any vital record specifications, and is no longer retained by a vital record specification, the data sets are eligible for expiration processing.

To have DFSMSrmm automatically delete a vital record specification, set a delete date when you add or change vital record specification information.

To manually delete a vital record specification:

1. Select Option 4 (DELETE) on the Vital Record Specification Menu and press ENTER. DFSMSrmm displays the Delete Vital Record Specification panel, as Figure 62 on page 97 shows:

Panel Help		
EDGPV400	DFSMSrmm Delete Vital Record Specification	VRS deleted
Command ==>		
Specify one of the following:		
Data set mask . .		
Job name mask . .		
Volume serial . .	(May be generic)	
VRS name		

Figure 62. DFSMSrmm Delete Vital Record Specification Panel

2. Enter a data set name, a volume serial number, or a vital record specification name.

Note: You must specify a data set name before you can specify a job name. Press PF1 for help.

3. Press ENTER.

DFSMSrmm displays either a Delete Data Set VRS panel, a Delete Volume VRS panel, or a Delete Name VRS panel. If you are deleting a name vital record specification, DFSMSrmm does not check if other vital record specifications are still linking to it.

If you entered a vital record specification name, DFSMSrmm displays a Delete Name VRS panel containing information about the vital record specification you indicated. For example, DFSMSrmm displays a panel such as the one in Figure 63 on page 98:

Panel Help

EDGPV440 DFSMSrmm Delete Name VRS

Name : NXT1

Count : 2

Retention type : CYCLES

While cataloged : NO

Until expired : NO

Location : HOME

Number in location : 2

Next VRS in chain . : NXT2

Chain using . . : NEXT

Owner : WOODMW

Description . . . :

Delete date . . : 1999/365 (YYYY/DDD)

Press ENTER to DELETE the VRS, or END command to CANCEL.

Command ==>

Figure 63. DFSMSrmm Delete Name VRS Panel

Press ENTER to delete the vital record specification.

Chapter 6. Retaining and Moving Your Volumes

Use vital record specifications to define retention and movement policies for data sets and volumes. See “Chapter 5. Defining Vital Record Specifications” on page 79 for information about defining data set and volume vital record specifications. DFSMSrmm supports all the retention types described in “Types of Retention”. See “Defining Retention Policies for Data Sets and Volumes” on page 103 and “Defining Movement Policies for Volumes” on page 113 for more information.

DFSMSrmm processes retention and movement policies when you run vital record processing, which is part of DFSMSrmm inventory management. During vital record processing, DFSMSrmm identifies the volumes that should be retained or moved. DFSMSrmm marks the volume movement as pending during storage location management processing. You must confirm that the volume movement has been completed before DFSMSrmm can process the next move in the policy. You must make sure all pending volume moves, whether they are processed automatically or manually, are confirmed to DFSMSrmm. When volumes are entered into an automated tape library dataserver, DFSMSrmm automatically confirms the move for the volumes. You can confirm movement for volumes one at a time or perform global confirmation where all pending volume moves are confirmed at the same time. “Global Confirmation” on page 126 provides examples of global confirmation using the RMM CHANGEVOLUME subcommand.

DFSMSrmm creates a detailed report of data sets and volumes being retained, the vital record specification used, and the location where the data set or volume is retained. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on running inventory management and creating its associated reports. DFSMSrmm keeps the latest retention information for each data set in the control data set. You can request retention information using the RMM LISTDATASET subcommand described in “LISTDATASET: Displaying Information about a Data Set” on page 342.

Types of Retention

This section describes the types of retention you can use to define retention information in all vital record specification except volume vital record specification.

CYCLES

Define a vital record specification to indicate the minimum number of cycles or copies of a matching data set that should be kept. For CYCLES processing, DFSMSrmm sorts GDGs by generation and version numbers. For non-GDG data sets, DFSMSrmm considers each occurrence of a data set to be a cycle.

Example

To retain five versions of the GDG, NBHART.DATA, issue the command:

```
RMM ADDVRS DSNAME('NBHART.DATA') GDG CYCLES COUNT(5)
```

BYDAYSCYCLE

Define a vital record specification to indicate the minimum number of cycles or copies of a matching data set that should be kept. For BYDAYSCYCLES processing, DFSMSrmm considers all the data sets created in a single day to be a cycle.

Example

Define a vital record specification to indicate the minimum number of cycles or copies of a matching data set that should be kept. For BYDAYSCYCLE processing, DFSMSrmm sorts GDGs by generation and version numbers:

```
RMM ADDVRS DSNAME('NBHART.DATA') GDG BYDAYSCYCLE COUNT(5)
```

DAYS

Define a vital record specification to indicate a period of elapsed days. All data sets matching the data set name mask created during this period are retained.

Example

To retain data sets created in the last 30 days for user ID, NBHART, issue the command:

```
RMM ADDVRS DSNAME('NBHART.**') DAYS COUNT(30)
```

EXTRADAYS

Define a vital record specification to retain a data set for extra days beyond the normal retention period. VRSEL(NEW) must be specified to retain a data set for extra days.

Example

```
RMM ADDVRS DSNAME('NBHART.**') DAYS COUNT(30) NEXTVRS(DAYS5)
RMM ADDVRS NAME(DAYS5) EXTRADAYS COUNT(5) LOCATION(HOME)
```

LASTREFERENCEDAYS

Define a vital record specification to ask that DFSMSrmm retain all copies of the data set based on the number of days since the data set was last read or written.

Example

To retain data sets for user ID NBHART, based on the number of elapsed days since the data set was last read or written, issue the command:

```
RMM ADDVRS DSNAME('NBHART.**') LASTREFERENCEDAYS COUNT(30)
```

WHILECATALOG

You also can request that data sets be retained only while they are cataloged. For a single data set that spans multiple volumes, DFSMSrmm retains all other volumes on which the data set is still cataloged. However, if DFSMSrmm releases the data set during normal expiration processing, DFSMSrmm also releases all volumes of a multi-volume data set. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information about defining the parmlib OPTION UNCATALOG operand to control how DFSMSrmm uncatalogs data sets.

Data sets created by long running batch jobs might get opened before, but cataloged after a run of inventory management. To protect these data sets and retain the volume in its current location, set a catalog retention period by using the parmlib OPTION CATRETPD option. Setting the parmlib OPTION CATRETPD(12) RETPD(0) MAXRETPD(0) ensures retention is controlled by DFSMSrmm vital record specifications only.

Example

To request that all data sets be retained while they are cataloged, issue the command:

```
RMM ADDVRS DSNAME('**') WHILECATALOG
```

UNTILEXPIRED

You can use vital record specification policies to retain a volume in a location as long as the volume expiration date has not been reached. The volume expiration date corresponds to the latest of all data sets written on the volume. For example, a volume expiration date of September 1, 1994 is higher than an expiration date of September 1, 1993.

When multiple policies are defined for a data set, all conditions must be true for the volume to be retained. The data set is not retained by a vital record specification that specifies WHILECATALOG and UNTILEXPIRED, if:

- The data set is uncataloged or
- The data set's volume expiration has been reached or
- The retention amount has elapsed

When you request that the data set be retained until it expires, DFSMSrmm releases the volume after the expiration date has been reached, regardless of the overall retention amount or catalog status.

Example

To retain all data sets until the volume expiration date has been reached, issue the command:

```
RMM ADDVRS DSNAME('**') UNTILEXPIRED
```

By Data Set Open Status

Define a vital record specification to set a policy for data sets that are left open by a system failure or closed as a result of the abnormal end of a task. Defining vital record specifications to manage these data sets allows you to define a specific policy for these data sets. If you do not define a policy, these data sets are managed by the vital record specification to which they match using the data set and job name, SMS management class, or management value.

Example

Open data sets might have been left open by a system failure or might be in use during DFSMSrmm inventory management. To retain data sets left open by a system failure or that are in use, issue the command:

```
RMM ADDVRS DSNAME('OPEN') LASTREFERENCEDAYS COUNT(2)
```

Example

To retain data sets that were closed as a result of an abnormal end in a task: issue the command:

```
RMM ADDVRS DSNAME('ABEND') CYCLES COUNT(1)
```

By Job Name

Define a vital record specification to retain a volume based on the job name that created the data set. DFSMSrmm checks the job name in the vital record specification against the creating job name defined for each data set that matches the data set mask. The creating job name is the name of the job that created the data set and is defined to DFSMSrmm using the RMM ADDDATASET subcommand or DFSMSrmm ISPF dialog. DFSMSrmm uses the job name to retain the data set if a matching job name is found. If no creating job name is known for a data set, DFSMSrmm uses the volume job name to retain the data set. When no creating job name is known for a data set, the data set cannot be retained by vital record specification definitions that specify a job name mask.

Example

To retain all data sets created by the CHECK job, issue the command:

```
RMM ADDVRS DSNAME('**') JOBNAME(CHECK)
```

By Specific Volume Serial Number

Define a vital record specification to retain a volume for a certain number of days.

Example

To retain volume K00111 for 100 days in a storage location, issue the command:

```
RMM ADDVRS VOLUME(K00111) COUNT(100) LOCATION(REMOTE)
```

By Generic Volume Serial Number

Define a vital record specification to ask that DFSMSrmm retain one or more volumes that match a generic prefix.

Example

To retain the latest 100 volumes matching the generic volume serial number, K00, in a storage location, issue the command:

```
RMM ADDVRS VOLUME(K00*) COUNT(100) LOCATION(REMOTE)
```

By Specific Date

Define a deletion date for a vital record specification. When that date is reached, DFSMSrmm deletes the vital record specification and all data sets and volumes that match to the vital record specification become eligible for release processing.

Example

To retain the data set, 'MIKES.DONT.DELETE.UNTIL.BTHDAY', until either the deletion date 1998/361 is reached or the data set is uncataloged, issue the command:

```
RMM ADDVRS DSNAME('MIKES.DONT.DELETE.UNTIL.BTHDAY') COUNT -
  DELETEDATE(1998/361) LOCATION(REMOTE) STORENUMBER -
  WHILECATALOG
```

Defining Retention Policies for Data Sets and Volumes

You define vital record specifications to set retention policies for data sets or volumes. Data set vital record specifications apply to the volume on which the data set resides. Volume vital record specifications assume nothing about data sets on the volume.

The retention period you enter in a vital record specification overrides the expiration date set for a data set or volume, and can be used to extend the time a data set or volume is retained. You can set retention policies for data sets and volumes using the RMM ADDVRS subcommand or the DFSMSrmm ISPF dialog. You can define a vital record specification to cover a single data set and volume or use data set name masks and generic volume serial numbers to define vital record specifications to cover multiple data sets and volumes.

DFSMSrmm sets an expiration date for each volume defined to it, using one of the following:

- User-specified JCL expiration date for a data set on the volume, not to exceed the maximum retention period (MAXRETPD) set by your installation in parmlib member EDGRMMxx.
- Default retention period for data sets and volumes defined by your installation, if no retention period or expiration date was set in user-specified JCL for a data set on the volume
- Expiration date or retention period specified by a user for a volume when manually requesting a scratch volume or manually adding or changing information about the volume to DFSMSrmm. This value cannot exceed the maximum retention period (MAXRETPD) set by your installation in parmlib member EDGRMMxx.
- The DFSMSrmm EDGUX100 installation exit. Using DFSMSrmm installation exit EDGUX100, you can assign a vital record specification management value based on special JCL specified expiration dates.

The expiration dates 1999/365 and 1999/366 mean "never-scratch" dates when you specify the DFSMSrmm EDGRMMxx parmlib OPTION command with the MAXRETPD(NOLIMIT) operand. Although DFSMSrmm supports using these dates as meaning permanent retention, using these dates in your JCL does not prevent DFSMSrmm from allowing a volume to be expired, returned to scratch status, or written over.

To manage volumes using these expiration dates to mean permanent retention, you must ensure that DFSMSrmm processing does not override the "never-scratch" dates. Here are some examples where DFSMSrmm processing ignores the expiration date. For example, use the EXPIRYDATEIGNORE operand in any vital record specification and DFSMSrmm ignores expiration dates, even "never-scratch" dates. Also when you issue the DELETEVOLUME subcommand with the RELEASE operand, you are requesting that DFSMSrmm release the volume no matter what the expiration date is. Note also that DFSMSrmm does not consider the expiration date when determining if a file can be overwritten. DFSMSrmm looks at the parmlib OPTION MASTEROVERWRITE operand and parmlib VLPOOL EXPDTCHECK operand to determine if a file can be overwritten.

Defining Policies for Management Class and Management Value

You can define policies to manage tape data sets using SMS management class names, or with installation exit assigned management value, based on special expiration dates supplied in JCL.

A management class name or management value can be one to eight alphanumeric characters, beginning with an alphabetic character, and must follow standard MVS data set naming conventions. This name must be a single qualifier. For example, you can define the management class name M99000 for all data sets with the special date 99000.

Your installation must select the management class names and the management values to be used in defining policies. Then you define vital record specifications with data set name masks using the management class names and vital record specification management values. For system-managed data sets, you define a management class name. Management class names are assigned by your installation using ACS routines. For non-system-managed data sets, or for data sets that do not belong to a management class but need to be managed by special JCL-specified expiration dates, you define a vital record specification management value. Management values are assigned to data sets by your installation using the DFSMSrmm-supplied EDGUX100 installation exit as described in *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

Figure 64 defines a policy using a management class name to retain a data set with the special date 99000. The management class name M99000 is assigned by your installation through an ACS routine. The WHILECATALOG operand indicates the data set is to be retained while it is cataloged.

```
ADDVRS DSNAME('M99000') WHILECATALOG
```

Figure 64. Retaining a Data Set Using Management Class

Figure 65 defines a policy using a vital record specification management value to retain a data set with the special date 99000. The vital record specification management value D99000 is assigned by your installation in an installation exit. The policy also states that all data sets are to be retained under catalog control.

```
ADDVRS DSNAME('D99000') WHILECATALOG
```

Figure 65. Retaining a Data Set Using Management Value

You can also use a data set name mask to define a vital record specification that matches several management class names or vital record specification management values. For example, you could use the data set name mask M9* to define a vital record specification that covers any special dates from 98001 through 99366 that have been assigned a management class name or vital record specification management value. This data set name mask must be a single qualifier.

After you define vital record specifications, using either a management class name or a management value, DFSMSrmm selects the best matching vital record specification for a data set during inventory management vital record processing. In making this selection, DFSMSrmm selects a vital record specification management value only if no match to management class or data set name is possible.

Combining Retention Policies in a Vital Record Specification

If you use VRSEL(NEW), DFSMSrmm combines retention policies when both a primary and secondary vital record specification match to a data set. When one vital record specification that uses the data set name mask '*.**' matches to a data set and another vital record specification with a management class name vital record specification management value, also matches, DFSMSrmm tries to combine the retention policies. DFSMSrmm checks to see if merging of WHILECATALOG should be done. If you use the data set name mask '*.**' in a data set vital record specification that does not include UNTILEXPIRED, then the vital record specification management value is not applied.

You can specify combinations of retention policies on a single vital record specification. When you specify multiple options on a retention policy, all the conditions must be true for the data set to be retained. In this example, the data set is retained until the vital record specification is deleted on 1995/361 or until the data set is uncataloged.

```
RMM ADDVRS DSNAME('RTEAM.HAS.FUN') COUNT -  
    DELETEDATE(1995/361) LOCATION(REMOTE) STORENUMBER -  
    WHILECATALOG
```

You can request that a data set is retained until the volume expiration date is reached, while a data set is cataloged, or the combination of expiration date and catalog status. You identify the retention policy when you define a vital record specification with the UNTILEXPIRED or WHILECATALOG operands.

When you specify UNTILEXPIRED only, DFSMSrmm retains the data set as long as the volume expiration date has not been reached. When you specify WHILECATALOG only, DFSMSrmm retains the data set as long as the data set remains cataloged. When you specify UNTILEXPIRED and WHILECATALOG, DFSMSrmm retains the data set only when both conditions are true.

Figure 66 shows a single vital record specification defined with both UNTILEXPIRED and WHILECATALOG. Data sets are retained as long as both conditions are true.

```
RMM ADDVRS DSNAME('*.**') COUNT(99999) UNTILEXPIRED WHILECATALOG
```

Figure 66. Using UNTILEXPIRED and WHILECATALOG on a Single Vital Record Specification

When your installation is using VRSEL(NEW), UNTILEXPIRED can be used with all retention types.

Merging Retention Policies

When you are using VRSEL(OLD), and define a single vital record specification with multiple retention policies for a data set or volume, DFSMSrmm retains a volume only when all the retention conditions are true. The exception to this is when a data set matches to a data set name vital record specification with the UNTILEXPIRED operand and a vital record specification management value vital record specification with the WHILECATALOG operand. When this match occurs, only one of the two conditions needs to be true for a volume to be retained.

```
RMM ADDVRS DSNAME('A.**') UNTILEXPIRED
RMM ADDVRS DSNAME('D99000') WHILECATALOG
```

Figure 67. Merging UNTILEXPIRED with the WHILECATALOG Retention Policy

Defining System-Wide Vital Record Specifications

You can define system-wide retention policies for all data sets not covered by other vital record specifications. If you have no other vital record specifications, define a single vital record specification with a data set name mask of `'**'` to establish a system-wide default. When no other vital record specifications match more specifically, then DFSMSrmm uses the `'**'` vital record specification to manage the data set. With `VRSEL(NEW)`, you can have two vital record specifications match to a volume.

If you define a vital record specification with the `'**'` data set name mask and specify `VRSEL(NEW)`, DFSMSrmm only uses one vital record specification to retain the data set. If you define a vital record specification with the `'*.**'` data set name mask and use `VRSEL(NEW)`, DFSMSrmm treats the vital record specification with the `'*.**'` data set name mask as the primary vital record specification and looks for a secondary vital record specification. If you use vital record specification management values, only the `'**'` data set name mask can be used to specify system-wide retention values.

DFSMSrmm Retention Dates

DFSMSrmm uses the retention information in a vital record specification subchain to calculate the retention date. The data set retention date is the date when a data set is no longer retained by the current vital record specification subchain. If only a single vital record specification retains a data set, the retention date is the date when the data set is no longer retained by a vital record specification and is eligible for expiration processing. DFSMSrmm calculates a retention date for each data set on a volume. DFSMSrmm then uses the highest retention date of all the data sets on a volume retained by a vital record specification to calculate the volume retention date. Use the `RMM LISTDATASET` or the `RMM LISTVOLUME` commands or DFSMSrmm ISPF dialog to display retention dates. See “Chapter 5. Defining Vital Record Specifications” on page 79 for information about how vital record specifications are defined and how policies are applied to data sets and volumes.

When vital record specifications are linked using `AND`, DFSMSrmm calculates the retention date based on the first vital record specification in the chain linked using `AND`. This means that the order you link the vital record specifications determines which retention date format DFSMSrmm calculates.

You control the way DFSMSrmm processes the retention information by specifying the `UNTILEXPIRED` operand on the vital record specification. Using `UNTILEXPIRED` is a way to combine retention policies so that DFSMSrmm uses the earliest possible date to calculate the retention date.

If only one vital record specification matches the data set, DFSMSrmm uses that vital record specification to retain the data set. If both a primary and secondary vital record specification match the data set name, and `UNTILEXPIRED` is not specified,

DFSMSrmm uses the vital record specification where the data set's point in time status matches the retention information in the vital record specification. This means that only one vital record specification retains a data set at a time, even if both vital record specifications match the data set. So, if the primary vital record specification matches, DFSMSrmm retains the data set using the primary vital record specification.

When both primary and secondary vital record specifications match the data set name, and you have specified UNTILEXPIRED, you are asking DFSMSrmm to combine the retention information in both vital record specifications. DFSMSrmm combines the retention information based on whether UNTILEXPIRED is defined in either the primary or secondary vital record specification or both. DFSMSrmm uses both vital record specifications to determine how to retain the data set. DFSMSrmm calculates the retention date based on the vital record specification that has the earliest date to stop retaining the data set.

Retention Date Format

DFSMSrmm displays retention date in date format as shown in Table 8 or with DFSMSrmm special date formats.

Table 8. Date Format Displayed

Language	Format	Example
American	mm/dd/yyyy	12/15/1994
European	dd/mm/yyyy	15/12/1994
Iso	yyyy/mm/dd	1994/12/15
Julian	yyyy/ddd	1994/349

where mm is the two digit number for the month, dd is the two digit number for the day, ddd is the three digit number for the day, and yyyy is the four digit number for the year. DFSMSrmm uses a null date for volumes that are never retained by vital record specifications.

Special Cycles Date Format

Special cycles format is CYCL/cccc where ccccc is the number of cycles to be retained. You define the number of cycles to be retained with the COUNT and BYDAYSCYCLE or CYCLES operands in the RMM ADDVRS subcommand.

You can specify the retention amount using COUNT(nnnnn), where nnnnn is a number in the range 1 to 99999. You can use COUNT with CYCLES to specify the number of cycles a data set should be retained. For example, if you code CYCLES COUNT(5) in a vital record specification, DFSMSrmm displays the special cycles format CYCL/00005.

Special Catalog Date Format

Special catalog format is WHILECATLG and is displayed when the WHILECATALOG operand is specified in the RMM ADDVRS subcommand.

Special CATRETPD Date Format

The special CATRETPD date format is CATRETPD which is the minimum catalog retention period. CATRETPD is set for a data set retained by a vital record specification with the WHILECATALOG operand when the data set is not cataloged and the CATRETPD time period has not passed. In this example without using the RMM ADDVRS CATRETPD(12) operand to retain the data set created at 8:00 AM, DFSMSrmm does not retain the data set because it was not cataloged at 12:00 PM when DFSMSrmm inventory management was run.

8:00 AM	12:00 PM	2:00 PM	7:00 PM
data set created	HSKP starts	HSKP ends	step or job ends data set cataloged

In this example using the RMM ADDVRS CATRETPD(12) operand to retain the data set created at 8:00 AM, DFSMSrmm ignores the fact that the data set is not cataloged for 12 hours until 8 AM. DFSMSrmm retains the data set rather than releasing it at 7:00 PM when DFSMSrmm inventory management is run.

8:00 AM	12:00 PM	2:00 PM	7:00 PM
data set created	HSKP starts	HSKP ends	step or job ends data set cataloged

How DFSMSrmm Calculates the Retention Date

DFSMSrmm calculates the retention date for data sets and volumes based on a combination of the RMM ADDVRS COUNT operand value and the retention type operands. The retention type operands are BYDAYSCYCLE, CYCLES, DAYS, LASTREFERENCEDAYS, UNTILEXPIRED, WHILECATALOG, and EXTRADAYS.

You can specify the COUNT operand value as a number from 1 to 99999. The COUNT operand value 99999 means to retain a data set or volume forever.

You can use the RMM ADDVRS RELEASE(EXPIRYDATEIGNORE) or RMM ADDVRS RELEASE(SCRATCHIMMEDIATE) operands to set the expiration date when you also the DFSMSrmm parmlib OPTION VRSEL(NEW) operand. When the only release action pending for a volume is to return to scratch status, you can override the expiration date set in JCL. Table 9 on page 109 shows the combinations of retention types, retention days or cycles that DFSMSrmm uses to calculate the retention date, and the way DFSMSrmm displays the retention date.

Data Set Retention Date

DFSMSrmm uses the earliest date for a data set as the retention date for the data set. DFSMSrmm sets the data set retention date using the current vital record subchain as follows:

1. CATRETPD when WHILECATALOG is specified and the data set is not cataloged and is created within the CATRETPD time period.
2. The earliest actual date, except for dates with special meaning. For example, dates with special meaning like 99365 and 99366 are specified to retain a volume permanently.
 - a. The date based on the date a data set is created or last referenced when the DAYS or LASTREFERENCEDAYS retention type is specified.

- b. The date based on the date the name vital record specification starts to retain the data set when the EXTRADAYS retention type is specified.
 - c. The volume expiration date if the UNTILEXPIRED retention type is specified.
 - d. The vital record specification deletion date if DELETEDATE() is specified.
 - e. The date when a volume is no longer retained by a vital record specification because the RELEASE(EXPIRYDATEIGNORE) is specified.
3. WHILECATLG if WHILECATALOG specified.
 4. 'CYCL/cccc' if BYDAYSCYCLE or CYCLES specified.
 5. '1999/365' which specifies that the data set is retained forever.

Note: DFSMSrmm uses the retention date that is the earliest when a data set matches to both a primary and secondary vital record specification, is still eligible for retention, and the primary vital record specification includes the UNTILEXPIRED retention type.

Volume Retention Date

DFSMSrmm uses the latest date for all the data sets on the volume as the retention date for the volume. DFSMSrmm sets the volume retention date in the following order:

1. 1999/366
2. 1999/365
3. CYCL/cccc
4. WHILECATLG
5. A date in the format selected by your installation
6. CATRETPD

Table 9 and Table 10 on page 111 describe how DFSMSrmm calculates retention date.

Table 9. DFSMSrmm Retention Date Calculation by COUNT from 1 through 99998

If you have Retention Type:	RMM Calculates Retention Date as:	And Displays Retention Date as:
CYCLES	Special cycles date format ¹	CYCL/cccc
DAYS	COUNT plus the create date ¹	Date format specified by your installation
EXTRADAYS (XD)	COUNT plus the date ¹⁵	Date format specified by your installation
LASTREF	COUNT plus the last reference date ¹	Date format specified by your installation
CYCLES + WC	Special catalog date format ³	WHILECATLG
DAYS + WC	COUNT plus the create date ¹	Date format specified by your installation
LASTREF + WC	COUNT plus the last reference date ¹	Date format specified by your installation
CYCLES + UEX	Volume expiration date ¹	Date format specified by your installation
DAYS + UEX	Lower of volume expiration date and COUNT plus create date ¹	Date format specified by your installation
LASTREF + UEX	Lower of volume expiration date and COUNT plus date last referenced ¹	Date format specified by your installation

Table 9. DFSMSrmm Retention Date Calculation by COUNT from 1 through 99998 (continued)

If you have Retention Type:	RMM Calculates Retention Date as:	And Displays Retention Date as:
CYCLES + WC + UEX	Volume expiration date ¹	Date format specified by your installation
DAYS + WC + UEX	Lower of volume expiration date and COUNT plus create date ¹	Date format specified by your installation
LASTREF + WC + UEX	Lower of volume expiration date and COUNT plus date last referenced ¹	Date format specified by your installation
(CYCLES + UEX) and (MV = WC) ²	Special catalog date format ³	WHILECATLG
(DAYS + UEX) and (MV = WC) ²	COUNT plus the create date ¹	Date format specified by your installation
(LASTREF + UEX) and (MV = WC) ²	COUNT plus the last reference date ¹	Date format specified by your installation
(CYCLES + UEX) and (MV/MC = WC) ²	Special catalog date format ³	WHILECATLG
(DAYS + UEX) and (MV/MC = WC) ²	COUNT + create date ¹	Date format specified by your installation
(LASTREF + UEX) and (MV/MC = WC) ²	COUNT + create date ¹	Date format specified by your installation
(DSN = UEX) and (MV/MC)	Calculates two dates. One date using the primary data set vital record specifications. One date using the secondary MV or MC vital record specification. DFSMSrmm uses the earliest of the two dates. ^{1, 4}	As determined by the vital record specification retention options.

Notes:

1. The vital record specification deletion date is used as long as it is not 1999/365 and is lower than the retention date calculated. DFSMSrmm calculates the deletion date by using the earlier of the vital record specification currently retaining the data set or the first vital record specification in the vital record specification chain.
2. Management value specifies WHILECATALOG and VRSEL(OLD) is in use. Also the data set is cataloged.
3. The vital record specification deletion date is used as long as it is not 1999/365 and is lower than the retention date calculated. If a data set is not cataloged and is retained using the parmlib CATRETPD operand, DFSMSrmm sets the CATRETPD retention date.
4. VRSEL(NEW) must be in use. The retention date format can be a date or any of the special date formats.
5. DFSMSrmm uses the date when the subchain started to retain the data set as the retention date.

Legend

- CYCLES= CYCLES and BYDAYSCYCLE retention types
- DSN = Matching data set vital record specification
- LASTREF = LASTREFERENCE (LASTREF date is the last referenced date in the data set record and is checked each time DFSMSrmm vital record processing is run.)
- MC = Management class
- MV = Management value
- UEX = UNTILEXPIRED
- WC = WHILECATALOG
- XD = EXTRADAYS

Table 10 shows how DFSMSrmm calculates retention date for various retention types when you specify a COUNT(99999) on the DFSMSrmm ADDVRS subcommand. The COUNT(99999) on the DFSMSrmm ADDVRS subcommand indicates that DFSMSrmm retains all cycles of a data set.

Table 10. DFSMSrmm Retention Date Calculation by COUNT(99999)

If you have Retention Type:	RMM Calculates Retention Date as:	And Displays Retention Date as:
CYCLES	VRS delete date OR cycles date format ¹	Date format specified by your installation or CYCL/nnnnn
DAYS	VRS delete date	Date format specified by your installation
EXTRADAYS (XD)	VRS delete date	Date format specified by your installation
LASTREF	VRS delete date	Date format specified by your installation
CYCLES + WC	VRS delete date OR catalog date format ²	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
DAYS + WC	VRS delete date OR catalog date format ²	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
LASTREF + WC	VRS delete date OR catalog date format ²	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
CYCLES + UEX	Volume expiration date ³	Date format specified by your installation
DAYS + UEX	Volume expiration date ³	Date format specified by your installation
LASTREF + UEX	Volume expiration date ³	Date format specified by your installation
CYCLES + WC + UEX	Volume expiration date ³	Date format specified by your installation
DAYS + WC + UEX	Volume expiration date ³	Date format specified by your installation
LASTREF + WC + UEX	Volume expiration date ³	Date format specified by your installation
(CYCLES + UEX) and (MV/MC = WC) ⁴	Data set VRS delete date OR catalog date format ²	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
(DAYS + UEX) and (MV/MC = WC) ⁴	Data set VRS delete date OR catalog date format ²	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
(LASTREF + UEX) and (MV/MC = WC) ⁴	Data set VRS delete date OR catalog date format ²	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
(DSN = UEX) and (MV/MC)	Calculates two dates. One date using the primary data set vital record specifications. One date using the secondary MV or MC vital record specification. DFSMSrmm uses the earliest of the two dates. ^{2, 5}	As determined by the vital record specification retention options.

Table 10. DFSMSrmm Retention Date Calculation by COUNT(99999) (continued)

If you have Retention Type:	RMM Calculates Retention Date as:	And Displays Retention Date as:
-----------------------------	-----------------------------------	---------------------------------

Notes:

1. The vital record specification deletion date is used as long as it is not 1999/365. The special cycles date format is used when the vital record specification deletion date is 1999/365. The deletion date used is the earlier of the current vital record specification and the first vital record specification in the vital record specification chain.
2. The vital record specification deletion date is used as long as it is not 1999/365. The special catalog date format is used when the vital record specification deletion date is 1999/365. If the data set is not cataloged and CATRETPD retains the data set, DFSMSrmm uses the CATRETPD retention date. The deletion date used is the earlier of the current vital record specification and the first vital record specification in the vital record specification chain.
3. The vital record specification deletion date is used as long as it is not 1999/365. The deletion date used is the earlier of the current vital record specification and the first vital record specification in the vital record specification chain.
4. Management value specifies WHILECATALOG and VRSEL(OLD). Also the data set is cataloged.
5. VRSEL(NEW) must be in use. DFSMSrmm calculates retention date based on both vital record specifications and information in this table.

Legend

- CYCLES= CYCLES and BYDAYSCYCLE retention types
- DSN = Matching data set vital record specification
- LASTREF = LASTREFERENCE
LASTREF date is the last referenced date in the data set record and is checked each time DFSMSrmm vital record processing is run.
- MC = Management class
- MV = Management value
- UEX = UNTILEXPIRED
- WC = WHILECATALOG
- XD = EXTRADAYS

Retaining Data Sets in Closed-Cycle GDGs

A closed-cycle GDG is a set of data sets that uses the volumes from the oldest generation each time that a new generation is created. Closed-cycle GDGs are commonly used in installations where a scratch pool of volumes has not been established or where no tape management system exists.

Managing Closed-Cycle GDGs

Define vital record specifications to set policies to retain all volumes that contain any of the closed-cycle GDG generations that match the GDG base name, but only move those volumes actually used in each generation. Specify CYCLES COUNT(99999) to keep all generations and do not use the retain while cataloged option WHILECATALOG.

For example, you have a closed-cycle GDG limited to three generations, want to keep the most recent generation in a system-managed library, ATL1, the next most recent in storage location REMOTE, and retain all volumes for reuse as the data sets fluctuate in size. Figure 68 on page 113 shows sample vital record specifications to manage the closed-cycle GDG.


```
RMM ADDVRS DSNAME('SAMPLE.GDG') GDG CYCLES COUNT LOCATION(ATL1) -
STORENUMBER(1) NEXTVRS(N1)
RMM ADDVRS NAME(N1) STORENUMBER(1) LOCATION(REMOTE)
```

Figure 68. Managing Closed-Cycle GDGs

Converting from Closed-Cycle GDGs to Using Scratch Pools

With closed-cycle GDGs, you might use more volumes than you actually require. We recommend that you consider moving to a scratch-pool-based system to use your volumes more efficiently.

1. Update the JCL that your applications use for creating the generations of their data sets on tape.
 - Modify the JCL to remove any special utilities that might be used to catalog the closed cycles.
 - Remove any references to the oldest generation of the GDG.
 - Ensure that the DD statement for the new generation does not indicate any volumes. The system will request a non-specific tape mount.
2. Update the vital record specifications shown in Figure 68. You can now use retention while cataloged and can optionally use the maximum number of generations that you want to be retained.

Modify the vital record specifications as shown in Figure 69.

```
RMM ADDVRS DSNAME('SAMPLE.GDG') GDG CYCLES COUNT(3) LOCATION(ATL1) -
STORENUMBER(1) NEXTVRS(N1) WHILECATALOG
RMM ADDVRS NAME(N1) STORENUMBER(1) LOCATION(REMOTE)
```

Figure 69. Managing Volumes Using Scratch Pools

During the first inventory management run after making these changes DFSMSrmm considers all the unwanted older generation volumes for release and expiration processing.

Defining Movement Policies for Volumes

DFSMSrmm manages the movement of volumes between the removable media library and storage locations, as well as among libraries within the removable media library. When you define a vital record specification for a data set or volume, you say where it should be retained to override any location set for the volume. When you run storage location management processing, a DFSMSrmm inventory management function, DFSMSrmm identifies a volume move and automatically assigns the volume a shelf location if the volume is moving to a shelf-managed storage location.

A volume's starting location is also known as its home location. Home location is set when you first define the volume to DFSMSrmm.

If the home location is a system-managed tape library, you cannot use SHELF to move a volume to a location in a non-system-managed tape library using vital record specifications.

You can change a volume's home location without initiating a move request, by using the RMM CHANGEVOLUME subcommand with the HOME operand. You can also change a volume's home location to any other location other than a storage location by using the RMM CHANGEVOLUME subcommand with the LOCATION. This request, however, initiates a volume move.

If you want to move a volume from a system-managed tape library to a non-system-managed library, use the Change Volume Details panel or the CHANGEVOLUME subcommand. See "Manually Requesting Volume Moves" on page 118 for more information.

Moving between System-Managed Libraries

To move a volume from one system-managed library to another, specify an installation-defined library name as a destination in your vital record specification. You can produce a report after vital record processing has been run to determine which volumes must be ejected and moved. Use the RMM CHANGEVOLUME subcommand with the EJECT operand to get volumes ejected at the right time. See "CHANGEVOLUME: Changing Volume Information" on page 282 for more information on CHANGEVOLUME.

When you use library names to identify automated or manual tape library dataservers in the removable media library, DFSMSrmm validates these names by ensuring that they have been defined in the TCDB.

Managing the Contents of System-Managed Libraries

You can use DFSMSrmm vital records selection processing to identify volumes that can be moved to free up space in an automated tape library datsarver when:

- You find that the libraries are too full to accept new volumes
- No more scratch volumes are available
- You have no room for volumes that you wish to use

In the following examples it is assumed that you run vital records selection processing on a regular basis. Define vital record specifications for volumes that can be moved to free up space based on installation policies and information available from DFSMSrmm. The vital record specifications might differ from other vital record specifications you define in that they are in effect for a short period of time or only manage a subset of volumes or data sets. The vital record specifications must use more specific data set name filters to override existing vital record specifications. They should include information about how long data resides in a system-managed library and when it can be removed and returned. After you define the vital record specifications they take effect at the next run of vital records selection processing and can be deleted at any time so that the previous policies are in effect.

Moving Volumes Out of System-Managed Libraries

Volumes containing cataloged data sets that are referenced soon after creation but then rarely referenced again are candidates for moving out of the library to free up space.

Figure 70 shows how DFSMSrmm can be used to help you manage your library when it is nearing its capacity.

```
RMM ADDVRS DSNAME('**') LASTREFERENCEDAYS COUNT WHILECATALOG -  
    LOCATION STORENUMBER(10) NEXTVRS(L1)  
RMM ADDVRS NAME(L1) LOCATION(LOCAL) STORENUMBER
```

Figure 70. Moving Volumes out of a System-Managed Library

The vital record specifications shown in Figure 70 ask that a data set remains in its home location for ten days since it was last referenced. After ten days the volume is marked to be moved to the LOCAL storage location. When the data set is uncataloged the volume is returned to its home location and then eligible to return to scratch.

In Figure 70, LOCAL is identified as the on-site location to store volumes that have to be removed from the automated tape library dataserer. Any other storage location or a manual tape library dataserer could have been coded as a target location.

When vital records selection processing is run, and if the data set has not been referenced, the volume is marked for movement to the LOCAL storage location. After volumes are marked for movement, you eject the volumes and move them to the designated storage location.

In Figure 70, when the library is running out of free space, the volumes moving to the LOCAL storage location could be ejected. You could eject all of them or select just as many as you require to make room for new volumes. You can use the RMM SEARCHVOLUME subcommand or the DFSMSrmm ISPF dialog to identify volumes to be ejected. You can also use information in the DFSMSrmm extract data set to select volumes that can be moved.

Figure 70 also uses the LASTREFERENCEDAYS operand. This operand indicates that if the volume is referenced after it is ejected, vital records selection processing will mark the volume to be moved back to its home location. You can avoid this by coding DAYS or CYCLES instead of LASTREFERENCEDAYS. DAYS bases vital records selection on the data set creation date rather than the date the data set was last referenced. Once the volume is ejected, reference to the data can be supported without it being returned to the library by vital records processing.

If you have no tape drives outside of your system managed libraries you could use the support provided by DFSMSrmm and OAM volume-not-in-library processing to get volumes returned to the library when they are needed. Refer to *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information.

Rotating Volumes In and Out of the System-Managed Library

Figure 71 on page 116 shows vital record specifications coded for a data set that is updated on a regular but infrequent basis. You want the data set to be resident in a library when the volume is required, but at other times you want the volume removed from the automated tape library dataserer.

```
RMM ADDVRS DSNAME('USERA.MONTHLY.DATA') LASTREFERENCEDAYS COUNT(365) -
  LOCATION(LOCAL) STORENUMBER(27) NEXTVRS(N1)
RMM ADDVRS NAME(N1) LOCATION STORENUMBER(30)
```

Figure 71. Moving a Volume in and out of the System-managed Library

In Figure 71 data set USERA.MONTHLY.DATA is always stored in the LOCAL storage location for the first twenty seven days since it was last referenced, it is then returned to its home location ready for the monthly update. The job is not delayed because the volume is in the library when the job runs. The job references the data so it starts off the vital records selection process at the LOCAL store for the next 27 days. If the data set is not referenced for one year it will be eligible for expiration processing.

Implementing a Data Archive Process

Figure 72 shows how to implement a data archive process. You can use days since creation as the criteria for moving a volume out of the library. Data you know is rarely referenced can automatically be moved out of the library at a fixed interval after creation.

```
RMM ADDVRS DSNAME('HSMX.HMIGTAPE.**') DAYS COUNT -
  LOCATION STORENUMBER(90) NEXTVRS(L1)
RMM ADDVRS NAME(L1) LOCATION(LOCAL) STORENUMBER
```

Figure 72. Moving Data out of the System-managed Library at Fixed Intervals

In Figure 72, DFSMSShsm migration tapes are moved from the library 90 days after creation to an on-site store called LOCAL. They remain in the LOCAL storage location until recycled by DFSMSShsm.

Moving to Storage Locations

During inventory management, DFSMSRmm identifies when volumes must be moved to one or more storage locations based on vital record specification information. DFSMSRmm has two types of storage locations:

- Built-in storage locations
 - LOCAL
 - DISTANT
 - REMOTE
- Installation defined storage locations which are defined with the LOCDEF command definitions in the DFSMSRmm parmlib member.

When the move destination is a DFSMSRmm built-in storage location, DFSMSRmm automatically assigns bin numbers where the volumes are to be stored. When the volume is to move to an installation defined storage location, DFSMSRmm assigns bin numbers using the bin numbers your installation has defined for the storage location if the storage location is shelf managed.

There is a predefined priority for moving volumes that DFSMSRmm uses when there is a conflict in the move destination for a volume. For example, if two vital record specifications defined for a volume are processed simultaneously, and one requests that the volume be moved to the REMOTE storage location while the second requests that the volume be moved to the LOCAL storage location, DFSMSRmm

selects the REMOTE storage location. Your installation can change this priority by adding priority to the LOCDEF location definition or by adding the priority to the vital record specification.

Using Manual Move Control

You can override automatic processing and control volume movement manually by using the RMM CHANGEVOLUME subcommand with the MANUALMOVE operand. To return the volume to automatic movement control, use the RMM CHANGEVOLUME subcommand with the AUTOMOVE operand.

When you put a volume under manual move control, DFSMSrmm does not move the volume anywhere automatically, even when it expires and is pending release. Volume movement occurs only if you request it using the RMM CHANGEVOLUME subcommand with the LOCATION operand.

To allow release processing, you must remove the volume from manual move control unless the volume is in its home location. When a volume is in its home location, release processing is performed even if the volume is under manual move control.

You might use manual move control to keep volumes on-site even though they are flagged to be sent off-site for disaster recovery. To keep the volume on-site, or to request it be moved back to its home location, you could specify:

```
RMM CHANGEVOLUME volser MANUALMOVE LOCATION
```

Figure 73. Keeping Volumes On-site

When a volume is put under manual move control, any outstanding move is canceled. Moves can also be canceled by issuing the RMM CHANGEVOLUME command with the LOCATION operand. The operand LOCATION(HOME) is specified in Figure 73 to cancel any pending moves because the volume is in its home location.

You might use manual move control for volumes you create on one system and then send to other systems for processing. Define the other systems as locations using the parmlib LOCDEF command. When a volume is ready to be sent to the other system, you can confirm the volume move and put the volume under manual move control at the same time. For example, to send a volume to another system defined on a LOCDEF command as OTHER1, you can issue the command:

```
RMM CHANGEVOLUME volser LOCATION(OTHER1) CONFIRMMOVE MANUALMOVE
```

Figure 74. Sending a Volume to Another System

The MANUALMOVE operand shown in Figure 74 puts the volume under manual move control and cancels any outstanding moves. This prevents the volume from being moved automatically. The LOCATION operand sets the destination for volume. The CONFIRMMOVE operand shown in Figure 74 confirms that the volume move has completed. When the volume is returned from the other system, remove the volume from manual move control and confirm that the volume is back in its home location as shown in Figure 75 on page 118.

Figure 75. Returning a Volume from Another System

Manually Requesting Volume Moves

To manually request a volume move, define a new location name for the volume using the DFSMSrmm ISPF Change Volume panel or the RMM CHANGEVOLUME subcommand. You can also use the RMM SEARCHVOLUME subcommand to create a data set of executable move commands for a list of volumes.

By supplying a new location name, you can manually request that volumes be returned from a storage location, moved among system-managed libraries, or returned to a shelf location in a non-system-managed tape library. Supplying a location name other than a storage location name, sets the volume's new home location, which is where the volume is returned after vital record processing.

If a destination is already entered for a volume in transit, you can cancel the move using the RMM CHANGEVOLUME subcommand with the LOCATION operand specifying the current location. If the volume does not have a destination, but has simply been ejected from an automated or manual tape library dataset server, you can change its location name while the volume is still in transit.

You must confirm all moves as completed after they have been performed. See "Confirming Volume Movements to DFSMSrmm" on page 124 for more information.

You can also use RMM subcommands to indicate when volumes move to loan locations outside of the removable media library and storage locations. DFSMSrmm movement and inventory reports do not include reports for volumes residing in loan locations. To get this information, use the RMM TSO SEARCH subcommands to list volumes based on loan location information. See "Chapter 10. Requesting Information about Your Resources" on page 163 for information on creating lists for scratch tapes available for use, and building drop/ship lists for volumes being moved to and from locations.

Requesting a Single Volume Move

To manually request a move for a single volume, use the Change Volume Details panel:

1. Enter the volume's serial number on the Change Volume panel and press ENTER. DFSMSrmm displays the Change Volume Details panel, containing information defined to DFSMSrmm for the volume.

For example, DFSMSrmm displays a panel such as the one in Figure 76 on page 119⁷:

7. Import/Export support is available with APAR OW36342 or OW36343.

Panel Help

EDGPT310 DFSMSrmm Change Volume Details -

Command ==>

Volume : A00001

Media name : 3480

Storage group . . . :

Move mode : AUTO

Volume type : PHYSICAL

Retention period . . :

Status : SCRATCH

Description :

Data set name . . . : 'SHELF.REPORT.DS001'

Media type : *

Label : SL

Current version . . :

Required version . . :

Density : *

Recording format . . :

Compaction :

Attributes :

Availability :

Owner :

Assigned date . . . : 1998/253

Last read date . . . :

Security name . . . :

Classification . . . :

Account number . . . :

Jobname :

Loan location . . . :

Previous volume . . :

Volume access list :

User :

User :

User :

User :

User :

User :

Pool :

or

Rack : A00001

Location name . . . :

Destination :

Bin number :

In container :

Expiration date . . : 1999/015

Initialize : NO

Release actions:

Return to SCRATCH pool . . :

Replace volume :

Return to owner :

Initialize volume :

Erase volume :

Notify owner :

Expiry date ignore . . . : NO

Scratch immediate . . . : NO

Owner access :

Assigned time : 00:27:17

Last write date :

MVS use : YES

VM use : NO

Last changed by : *HKP

Next volume :

Access :

User :

User :

User :

User :

User :

User :

More: +

YYYY/DDD

Enter SCROLL commands for more volume information, END command to CANCEL.

Figure 76. DFSMSrmm Change Volume Details Panel

2. Enter a value in the Location Name field. Press PF1 or use the HELP command for more information.
3. Press ENTER to update the information.

You can also issue the RMM CHANGEVOLUME subcommand with a volume serial number and the LOCATION operand to supply a new location name.

For example, to request that volume A00001 currently residing in manual tape library dataserver, LIB1, be moved to LIB2, use:

```
RMM CHANGEVOLUME A00001 LOCATION(LIB2)
```

You can also use CHANGEVOLUME with the CONFIRMMOVE operand to confirm that a move has been completed. See “CHANGEVOLUME: Changing Volume Information” on page 282 for more information on CHANGEVOLUME.

Moving Volumes to a System-Managed Tape Library

To request that one or more volumes be moved to an automated or manual tape library dataserer, enter a library name as the new location on the Change Volume panel or on the RMM CHANGEVOLUME subcommand. For example, to use the RMM CHANGEVOLUME subcommand to move a volume to LIB2, enter:

```
RMM CHANGEVOLUME volser LOCATION(LIB2)
```

Or, to use the Change Volume Details panel to return a volume from a storage location to its home location in LIB2, enter the following in the Location name field:

```
Location name ===> HOME
```

When you give the name of a system-managed library as a new location name for a volume, DFSMSrmm first validates the volume serial number and library name to ensure it is defined in the TCDB. You can supply this new location name either before or after you physically move the volume to its new location.

DFSMSrmm manages volume moves to a system-managed library as follows:

- For a move to an automated tape library dataserer, the operator must move the volume to the automated tape library dataserer. During cartridge entry processing to an automated tape library dataserer, DFSMSrmm checks:
 - Only the supported label types (SL, AL) are used
 - Internal and external volume serial numbers match
 - No duplicate volume serial numbers or rack numbers are allowed

DFSMSrmm automatically confirms the volume move as complete. If DFSMSrmm was inactive during cartridge entry processing, you must manually confirm that the volume was moved.

- For a move to a manual tape library dataserer, DFSMSrmm requests that the TCDB be updated with the volume's new location. No physical move is required, because the rack number of the shelf location where the volume is stored remains the same.

All volumes should be defined to DFSMSrmm prior to moving them to a system-managed library. However, if you enter a volume in a system-managed library prior to entering a location name, DFSMSrmm updates the volume information in its control data set with the new library name and type, and with the volume entry status showing that the volume resides in the library.

If you enter a volume into a different system-managed library than that already defined for the volume, DFSMSrmm corrects the library name in the control data set. If you enter a volume into a system-managed library and the destination specified for the volume is not the library in which you are entering the volume, DFSMSrmm does not update the volume information. DFSMSrmm rejects the entry request and the volume is ejected.

Ejecting Volumes from a System-Managed Tape Library

For a volume in a automated tape library dataserer, except for a logical volume residing in a VTS, the volume is physically ejected and must be moved to its

destination by the operator⁸. DFSMSrmm automatically records the move as started when the volume has been ejected, and records the volume as being in transit until you confirm the move as complete.

For a manual tape library dataserer, no physical eject takes place. DFSMSrmm updates the TCDB with the volume's correct location. If the volume is being moved to an automated tape library dataserer, to a storage location, or to a location outside the removable media library, the operator must move the volume.

Once a volume is ejected from a system-managed library, DFSMSrmm records it as being in transit until you confirm the move as completed using the CHANGEVOLUME subcommand or the Volume Action Status panel. When a volume is ejected, DFSMSrmm deletes the volume record from the TCDB.

DFSMSrmm does not automatically eject volumes from a system-managed tape library as part of inventory management processing. To request that a volume is ejected from a system-managed tape library, use the DFSMSrmm Change Volume Details panel to change the location for a volume residing in a system-managed library or use the RMM CHANGEVOLUME subcommand to change the location or issue an eject request. Figure 77 shows how to use the CHANGEVOLUME subcommand to change a volume's location to its shelf location. The change to the SHELF causes the volume to be ejected from a system-managed tape library.

```
RMM CHANGEVOLUME volser LOCATION(SHELF)
```

Figure 77. Ejecting a Volume from a System-managed Library

You could also use the Change Volume panel to return a volume to its home location by entering HOME in the Location name field:

```
Location name ==> HOME
```

You could also set a loan location value for a volume, using the ISPF Change Volume panel, or when you issue:

```
RMM CHANGEVOLUME volser LOANLOC(loan_location)
```

You can use the EJECT operand on the CHANGEVOLUME and DELETEVOLUME subcommands to indicate where you want volumes to be ejected. DFSMSrmm ejects volumes to the convenience output station unless you request otherwise. You can use the DELETEVOLUME subcommand with the NOEJECT operand to prevent ejects.

You can also eject a volume when you release a volume using the Release Volumes panel and you issue either Remove or Force as a release type, or when you issue:

```
RMM DELETEVOLUME volser REMOVE
```

or

```
RMM DELETEVOLUME volser FORCE
```

To build a list of CHANGEVOLUME subcommands to eject volumes moving from LIB1, specify:

8. Import/Export support is available with APAR OW36342 or OW36343.

```
RMM SEARCHVOLUME VOLUME(*) LOCATION(LIB1) -
    DESTINATION(*) INTRANSIT(NO) -
    CLIST('RMM CHANGEVOLUME ', ' EJECT(BULK)')
```

Moving Volumes Between Pools

You might find it necessary to move volumes from one pool to another. For example, you might move volumes to eliminate an old pool, to consolidate space as your installation's removable media library grows, or to change a volume's external label.

If the pools between which you are moving volumes reside in different libraries, you might need to move volumes between libraries at the same time you redefine a new shelf location for the volume.

To view the pool IDs defined for your installation, type `CONTROL VLPOOLS` from the command or option line of any panel. In the TSO environment, use the `LISTCONTROL VLPOOL` subcommand.

Moving between pools is not supported in automated tape library dataservers.

To change the shelf location for a volume being moved between pools, do one of the following:

- Enter a new pool ID in the Pool field, or a new rack number in the Rack field of the Change Volume Details panel, and press ENTER.

or

- Issue either:

```
RMM CHANGEVOLUME volser POOL(pool_ID)
```

or

```
RMM CHANGEVOLUME volser RACK(rack_number)
```

where:

pool ID is one to five characters followed by an * and must be defined by your installation.

rack_number is the rack number of the shelf location where you are moving the volume. It can be a full rack number of one to six alphanumeric characters, or a generic rack number of one to five alphanumeric characters followed by an asterisk. The rack number you use must correspond to an empty shelf location already defined to DFSMSrmm, and must match the media name of the volume you are moving.

After you redefine pool information for a volume, replace the volume's external label with a new label identifying the volume's new shelf location by a rack number.

Moving Volumes to Loan Locations

Volumes are not always kept in the removable media library or in designated storage locations, but are sometimes loaned out or retained by owners outside the library. To keep track of these locations, define a *loan location* to DFSMSrmm when you add or change volume information.

A loan location is any value that identifies where the volume can be found when it is stored outside your installation's removable media library, such as an owner name

or user ID, a department name, or an office number. A loan location value can be any value up to eight characters. You can change the loan location for a volume as often as needed.

To specify a loan location for a volume already defined to DFSMSrmm, either:

- Enter a value in the Loan Location field of page 2 of the Change Volume Details panel.
- or
- Use the RMM CHANGEVOLUME subcommand as shown in Figure 78 to request that volume VOL001 is being moved to an office in building 86 with an office number 201EE.

```
RMM CHANGEVOLUME VOL001 LOANLOC(86201EE)
```

Figure 78. Moving a Volume to a Loan Location

Figure 79 shows how to build an executable data set of CHANGEVOLUME commands supplying a loan location for several volumes already defined to DFSMSrmm.

```
RMM SEARCHVOLUME VOLUME(full_or_generic_volume_serial)-  
CLIST('RMM CHANGEVOLUME ', ' LOANLOC(loan_location)')
```

Figure 79. Supplying Loan Location Information for Volumes

where VOLUME is a generic volume serial number, and *loan location* is a value of one to eight characters.

Returning Volumes from Loan Locations

To return a volume from a loan location, the loan location value for the volume must be cleared.

If a volume assigned to a loan location is entered into a system-managed tape library, DFSMSrmm automatically clears the loan location value.

If a volume resides in a non-system-managed tape library, use the DFSMSrmm ISPF dialog or the RMM CHANGEVOLUME subcommand as shown in Figure 80 to clear the loan location value.

```
RMM CHANGEVOLUME volser LOANLOC('')
```

Figure 80. Returning a Volume from a Loan Location

See “CHANGEVOLUME: Changing Volume Information” on page 282 and “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information.

Moving Volumes from Storage Locations

If you want to move a volume in a storage location back into the library without performing vital record processing or storage location management processing, you can use the RMM CHANGEVOLUME subcommand. DFSMSrmm updates the control data set to reflect the new location.

Note: Vital records processing and storage location override any changes made manually. Be aware that during the next run of vital records or storage location processing, the volume might get marked for return to the storage location from which it was moved, unless you place the volume under manual move control.

Confirming Volume Movements to DFSMSrmm

Volume movement must be confirmed to DFSMSrmm.

DFSMSrmm performs the volume movement confirmation for volumes in a system-managed library by:

- Confirming the move when a volume is entered into an automated tape library dataserer.
- Confirming that a volume is ejected from a system-managed library. When a volume is ejected from a system-managed library, DFSMSrmm marks the volume as being in-transit until you confirm that the volume has completed its move to another location. For volumes moving from an automated or manual tape library dataserer, as directed by vital record processing, you must manually eject the volume to start the move. You can only confirm such a move after the eject has taken place.

You must confirm that movement is complete when moving volumes involves manual tasks external to DFSMSrmm, such as filling and transporting boxes to storage locations. You can confirm movement for a single volume at a time or you can perform global confirmation for many volumes at one time.

You might be required to confirm movement for a volume even though it is no longer retained by a vital record specification. If a volume has a move destination set, DFSMSrmm will not cancel the move. The move must be completed, and once it is confirmed, if the volume is not in its home location, the volume is identified for movement to return to its home location. You will then have to confirm the movement back to the home location.

If you want to avoid the move and move confirmation, you can cancel the move. DFSMSrmm then does not require the confirmation of the move and attempts to move the volume to its home location during the next run of DFSMSrmm inventory management.

Confirming Volume Moves into an Automated Tape Library

To confirm volume moves into an automated tape library dataserer, enter the volume into the automated tape library dataserer, and DFSMSrmm automatically confirms the move as completed.

Confirming Volume Moves for Other Locations

To confirm volume moves for other locations, you can use either the ISPF Volume Action Status panel or the RMM CHANGEVOLUME subcommand to confirm to DFSMSrmm that a move was performed. You can also use the Volume Action Status panel or the RMM SEARCHVOLUME subcommand to create a list of moves to be performed and confirmed, and use line operators from the list to confirm the moves.

Using the Volume Action Status Panel

To manually confirm one or more volume moves:

1. Select Option 8 (CONFIRM) on the Volume menu.

Note to Tape Librarians: You can bypass the Volume Menu by selecting Option 9 (CONFIRM) on the Librarian Menu.

2. Press ENTER. DFSMSrmm displays the Volume Action Status panel as shown in Figure 81:

Panel Help

EDGPC700 DFSMSrmm Volume Action Status Row 1 to 8 of 8

Command ==> Scroll ==> PAGE

The following line commands are valid: C, U and S

Dest-

S Action Location ination Move Type Status

ERASE PENDING

INIT PENDING

NOTIFY PENDING

REPLACE UNKNOWN

RETURN UNKNOWN

SCRATCH PENDING

C MOVE LIBRARY DISTANT NOTRTS PENDING

MOVE REMOTE LOCAL NOTRTS PENDING

***** Bottom of data *****

Figure 81. DFSMSrmm Volume Action Status Panel

3. Type C next to the type of move you want to confirm as shown in Figure 81. You can confirm as many types of moves as you want.
4. Press ENTER.
DFSMSrmm updates the DFSMSrmm control data set with the status and confirms all applicable moves during inventory management.

You can also use the Volume Action Status panel to request a list of volumes for a type of move, and confirm those moves from this list. To do this:

1. Type S next to the type of move for which you want to see a list of volumes. For example, to see a list of all volumes being moved from LIB1 to LIB2 enter:

		Dest-		
S	Action	Location	ination	Move Type Status

S	MOVE	LIB1	LIB2	NOTRTS Pending

2. Press ENTER.
DFSMSrmm returns a list such as the one in Figure 82 on page 126:

Panel Help

EDGPT610 DFSMSrmm Volume Action Summary List Row 1 to 4 of 4

Command ==> Scroll ==> PAGE

The following line commands are valid: C, CE, CI, CM, CN, CO, CR, E, L, and V

	Volume	Assigned	Expiration		Rack	Dest-	Tra-		
S	Serial	Owner	Date	Date	Location	Number	Action	ination	nsit
C	DMK000	ETZ001	1994/310	2019/092	LIB1	DMK000		LIB2	Y
	DMK001	ETZ001	1994/310	1994/315	LIB1	DMK001		LIB2	Y
	DMK002	ETZ001	1994/310	1994/360	LIB1	DMK002		LIB2	Y
	DMK003	ETZ001	1994/310	1994/335	LIB1	DMK003		LIB2	Y

***** Bottom of data *****

Figure 82. DFSMSrmm Volume Action Summary List

3. Use either the C or CM line operators against entries in the list to confirm outstanding moves.

Confirming Movement for a Specific Volume

Use the RMM CHANGEVOLUME subcommand with a volume serial number and the CONFIRMMOVE operand to confirm a single move. For example, to confirm a move for volume VOL001, enter:

```
RMM CHANGEVOLUME VOL001 CONFIRMMOVE
```

Global Confirmation

Use the RMM CHANGEVOLUME subcommand with an asterisk and the CONFIRMMOVE operand to confirm outstanding movement for multiple volumes.

When you use an asterisk, DFSMSrmm confirms moves for all volumes that have the outstanding moves you indicate on the CONFIRMMOVE operand during inventory management.

For example, to confirm all outstanding moves, enter:

```
RMM CHANGEVOLUME * CONFIRMMOVE(ALL,ALL)
```

You can also use CONFIRMMOVE with source and location values to identify which moves you are confirming. For example, to confirm all moves from a library with a library name of LIB1, enter:

```
RMM CHANGEVOLUME * CONFIRMMOVE(LIB1,ALL)
```

Or, to confirm all moves to a library with a library name of LIB2, enter:

```
RMM CHANGEVOLUME * CONFIRMMOVE(ALL,LIB2)
```

Or, to confirm all moves from LIB1 to LIB2, enter:

```
RMM CHANGEVOLUME * CONFIRMMOVE(LIB1,LIB2)
```

When you use the CONFIRMMOVE operand with an ALL value, the target libraries to which volumes with outstanding moves are moving must be defined on the system processing the CHANGEVOLUME subcommand. Any volumes moving to libraries that are not defined on the system are ignored and DFSMSrmm leaves those moves as pending.

You can also use CONFIRMMOVE with source and location values and the READYTOSCRATCH or NOTREADYTOSCRATCH operands. READYTOSCRATCH means the volume has no outstanding release actions so it can be confirmed and returned to scratch in one action. NOTREADYTOSCRATCH are private volumes and volumes with release actions other than return to scratch. Figure 83 shows how to confirm all moves from location REMOTE to location SHELF that are ready to scratch.

```
RMM CHANGEVOLUME * CONFIRMMOVE(REMOTE,SHELF,READYTOSCRATCH)
```

Figure 83. Confirming Volume Moves for Volumes Ready to Scratch

Use SEARCHVOLUME with the CLIST operand to create a data set of executable CHANGEVOLUME subcommands to confirm a list of volume moves.

You indicate the source and target locations identifying the move using the LOCATION and DESTINATION operands. For example, to build a list of CHANGEVOLUME subcommands to confirm all volume moves between LIB1 and LIB2, use the SEARCHVOLUME subcommand shown in Figure 84.

```
RMM SEARCHVOLUME VOLUME(*) LOCATION(LIB1) DESTINATION(LIB2) -  
    INTRANSIT CLIST('RMM CHANGEVOLUME ',' CONFIRMMOVE')
```

Figure 84. Identifying Locations and Destinations for Volumes

DFSMSrmm builds you a CLIST data set. You can edit this CLIST to remove any volumes that did not get moved, then you can run it at your convenience. If you use the MOVETYPE operand on the SEARCHVOLUME subcommand you can be selective about the volumes you process. For example, you could select only volumes that are ready to scratch.

To build a list of CHANGEVOLUME subcommands to confirm all replace actions, specify:

```
RMM SEARCHVOLUME VOLUME(*) ACTION(REPLACE)-  
    CLIST('RMM CHANGEVOLUME ',' CONFIRMRELEASE(REPLACE)')
```

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information on using SEARCHVOLUME with the CLIST operand. See “CLISTs of Executable Subcommands” on page 191 for more information on creating lists of executable subcommands.

Chapter 7. Defining Retention and Movement Policies

You use DFSMSrmm vital record specifications to define retention and movement policies for data sets and volumes. The policies can be a single vital record specification or vital record specification subchains linked with NEXTVRS or ANDVRS. The transition through the policy takes place as each subchain is processed. The transition is triggered by inventory management and point in time status.

Before You Start

This section summarizes the tasks that should be performed to implement retention and movement policies and points to a list of references for more information.

A Summary of the Tasks To Perform

1. Set installation-wide retention periods using the parmlib member EDGRMMxx OPTION command.
2. Set vital record processing options using the parmlib member EDGRMMxx OPTION command operands VRSCHANGE, VRSJOBNAME, VRSMIN, and VRSEL.
3. Define vital record specifications to retain and move data sets and volumes.
4. Define data sets and volumes for DFSMSrmm to manage.
5. Optionally run inventory management in trial run mode to see the effect of the policies you defined before they take affect.
6. Run DFSMSrmm inventory management vital record processing to process the policies you have defined.

Where to Find Information

We recommend you review the following references before defining policies to DFSMSrmm.

- “Chapter 5. Defining Vital Record Specifications” on page 79 and “Chapter 6. Retaining and Moving Your Volumes” on page 99 describe the retention and movement policies you can define with DFSMSrmm. The policies known as vital record specifications can be defined using the RMM ISPF dialog or the RMM TSO subcommand.
“Chapter 5. Defining Vital Record Specifications” on page 79 describes how to define vital record specifications using the RMM ISPF dialog.
- “Chapter 12. Using RMM TSO Subcommands” on page 221 describes the following RMM TSO subcommands: ADDVRS, DELETEVRS, LISTVRS, and SEARCHVRS.

Note: There is no CHANGEVRS subcommand. To change a policy using the RMM TSO subcommands, you must first use DELETEVRS to remove the vital record specification and then use ADDVRS to add the policy with the new values. We recommend using the RMM ISPF dialog because it provides a simulated CHANGEVRS capability which makes changing vital record specification values easier.

- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* Chapter 6 describes the DFSMSrmm parmlib member EDGRMMxx OPTION command operands. The operands CATRETPD, MAXRETPD, and RETPD are used to

specify retention periods. The operands VRSCCHANGE, VRSJOBNAME, VRSMIN, and VRSEL are used to control how DFSMSrmm processes retention and movement policies. The DFSMSrmm parmlib member EDGRMMxx LOCDEF command is used to define locations where volumes can be moved.

- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* Chapter 12 describes DFSMSrmm inventory management vital record processing that must be performed for retention and movement policies to take effect.
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* Chapter 10 provides vital record specification examples for retaining and moving DFSMSshm data sets.
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* Chapter 9 provides information about using the DFSMSrmm EDGUX100 exit to assign vital record management values to data sets.

Which Type of Vital Record Processing Should You Use?

You select the type of processing you want DFSMSrmm to perform by specifying the DFSMSrmm EDGRMMxx parmlib member OPTION VRSEL(OLD) or VRSEL(NEW) operand. If you use vital record specification retention and movement functions that are not supported by VRSEL(OLD), DFSMSrmm ignores the unsupported values, issues message EDG2227I, and continues to process the remaining policy. If you use VRSEL(NEW), DFSMSrmm processes any vital record specification functions that are available.

What is Your Level of Experience with DFSMSrmm?

Your level of experience with DFSMSrmm might determine which type of vital record processing you should use. Here are some suggestions for determining whether you should use VRSEL(OLD) or VRSEL(NEW) based on your experience with DFSMSrmm.

- If you are using DFSMSrmm in PROTECT mode, understand how the retention and movement policies using vital record specifications work, and are satisfied with the way DFSMSrmm is managing your data sets and volumes, use VRSEL(OLD).
- If you are using DFSMSrmm in PROTECT mode, understand how the retention and movement policies using vital record specifications work but see the need to use some of the new options, use VRSEL(NEW). Follow the steps described in “Migrating to Using VRSEL(NEW)” on page 132 and carefully implement needed functions.
- If you are in the migration process but are too close to going into PROTECT mode to consider new options, use VRSEL(OLD).
- If you are early in your migration to DFSMSrmm or are a new user, use the new conversion program EDGRVCLN to recreate the DFSMSrmm control data set and use VRSEL(NEW).

Selecting VRSEL(OLD) Processing

When you specify VRSEL(OLD), you can:

- Continue to use any vital record specifications you have defined previously and to expect that there is no change to those policies.
- Define vital record specifications with data set and volume names based on the matching order described in Table 5 on page 84.

- Define vital record specifications with only a single retention type that use the NEXTVRS operand to create chains that describe how to move data through several locations.
- Define vital record specification management values and management classes and use the DFSMSrmm EDGUX100 exit to assign these values so retention and movement policies can be applied to data sets and volumes.
- Use the UNTILEXPIRED operand so that DFSMSrmm merges the retention information from two vital record specifications to form a single policy.
- DFSMSrmm applies policies one vital record specification at a time when the retention criteria is true a data set.
- You must run DFSMSrmm inventory management vital record processing for DFSMSrmm to apply retention and movement policies to data sets and volumes. To release a volume, you must run inventory management expiration processing.
- You can use the current location where a data set resides instead of only considering the home location or destination when defining policies. This allows both automatic and command-driven movement to be used within an installation, particularly useful when the location to be managed needs to be managed to a threshold, such as for an Tape Library Dataserver

Selecting VRSEL(NEW) Processing

When you specify VRSEL(NEW), you can use the following additional functions not available when you use VRSEL(OLD).

- You can include retention information on a name vital record specification. If you chain vital record specifications using the RMM ADDVRS NEXTVRS or ANDVRS subcommand operands, each vital record specification can have a different retention type. If you use the ANDVRS operand, all the retention criteria you specify must be true for DFSMSrmm to apply a policy. If you define subchains in a vital record specification chain, DFSMSrmm applies the policies defined in a subchain independent of the rest of the chain.
- You can retain cycles of data sets by day, where a cycle is all copies of a data set created on the same day. This is an alternative to retaining cycles of data sets where a cycle is a single occurrence of a data set.
- You can use the current location where a data set resides instead of only considering the home location or destination when defining policies. This allows both automatic and command-driven movement to be used within an installation, particularly useful when the location to be managed needs to be managed to a threshold, such as for an Tape Library Dataserver.
- You can manage data set retention and movement separately. Use a vital record specification based on management value or management class name to define retention criteria; use a vital record specification based on data set name mask for defining movement.
- You can request that a volume expiration date be ignored when the data set is released from retention by the policy. See “Setting Release Actions for a Volume” on page 149 for information about release actions.
- You can request that volumes are returned to scratch status when the only release action is return to scratch.
- You can define vital record specification chains that have different retention types on each vital record specification in a chain.

- You can define vital record specification management values and management classes and use the DFSMSrmm EDGUX100 exit or your management class ACS routine to assign these values so retention policies can be applied to data sets and volumes.
- You run DFSMSrmm inventory management vital record processing for DFSMSrmm to apply retention and movement policies to data sets and volumes. To release a volume, you must run vital record processing and expiration processing. We recommend running inventory management vital record processing and expiration processing in a single job step.

Migrating to Using VRSEL(NEW)

We recommend using VRSEL(NEW) to:

- Have more flexibility in defining retention and movement policies
- Override a volume expiration date when a volume is dropped from vital record specification retention and the data set retained on it has the release option EXPIRYDATEIGNORE.
- Return a volume to scratch status in a single inventory management run

Before starting to use VRSEL(NEW), understand that your existing policies might be applied differently under VRSEL(NEW). You can perform these steps to avoid problems that might occur when the DFSMSrmm performs VRSEL(NEW) processing.

1. Before using VRSEL(NEW), back up your DFSMSrmm control data set using EDGBKUP or EDGHSKP.
2. Perform cleanup on the name vital record specifications by making sure that any retention information in them is correct. DFSMSrmm provides the EDGRVCLN REXX exec to report and clean up problems with name vital record specifications.
3. Run DFSMSrmm inventory management vital record processing so the DFSMSrmm control data set reflects the cleanup you have done.
4. Update the DFSMSrmm parmlib OPTION VRSEL(NEW) operand.
5. Make sure all systems sharing DFSMSrmm control data sets have the same parmlib options.
6. Run the inventory management VERIFY function against the control data set without introducing any of the new vital record specification functions. When you run VERIFY, changes are not actually made to the DFSMSrmm control data set so you can look at the results before any changes are made.
7. Inspect the inventory management VERIFY ACTIVITY file by looking at changes in matching vital record specification information, vital record status, and retention date. DFSMSrmm provides a sample job EDGJACTP that you can use with DFSORT to format and print fields in the ACTIVITY file. If you cannot cleanup to your satisfaction, you can revert to VRSEL(OLD) at this time.
8. Correct vital record specifications as needed to make sure that the policies you want are in place.
9. Continue running the VERIFY function and inspect the results until you get the results you expect using the new functions
10. Begin defining vital record specifications that include the new release options or the use of ANDVRS. We suggest starting slowly until you gain more experience with using the new vital record specification functions. We

recommend using the RMM ISPF dialog to add the new vital record specifications or to make changes to existing vital record specifications.

11. Repeat the process from step 7 on page 132, step 9 on page 132, and step 10 on page 132 until you are satisfied that the results are what you expect.
12. Run inventory management production run processing.
13. Check the vital records retention report to make sure that data sets and volumes are retained as you intended.

Chaining Retention and Movement Policies

A vital record specification chain is a data set or volume vital record specification and all of the name vital record specifications chained from it. A vital record specification subchain starts with a data set vital record specification, name vital record specification with retention information, or a group of vital records chained using the RMM ADDVRS ANDVRS operand. The subchain includes all the vital record specifications chained from the start of the chain. The subchain ends before the next vital record in the chain that contains retention information. The subchain in a chain is the part of the policy that DFSMSrmm is processing at a given time. Both a vital record specification chain and a vital record specification subchain can be one or more vital record specifications.

You can define retention and movement policies in a single vital record specification or in a chain of vital record specifications. Creating chains of vital record specifications allows you to define combinations of retention and movement criteria. You construct vital record specification chains by linking data set or name vital record specifications. The entire vital record specification chain is considered the policy for the data set.

To create a vital record specification chain, define a data set name vital record specification. Include the name of the next vital record specification in the chain using the RMM TSO ADDVRS subcommand NEXTVRS or ANDVRS operand.

- Use the NEXTVRS operand if you want DFSMSrmm to process each vital record specification in the chain separately.
- Use the ANDVRS operand if you want DFSMSrmm to process all the vital record specifications linked by the ANDVRS operand together.

When DFSMSrmm is processing a vital record specification chain, if retention defined in a subchain is true, then DFSMSrmm applies the policies in the subchain independently of the rest of the vital record specification chain. If any of the retention criteria in a subchain is not true, DFSMSrmm considers the next subchain, continuing until retention and movement are completed or the data set is dropped from vital record specification control.

What You Can Chain When You Specify VRSEL(OLD)

If you use the parmlib member EDGRMMxx OPTION VRSEL(OLD) operand, you can chain vital record specifications as follows:

- Data set vital record specifications contain the retention information for the data set.
- Any name vital record specification chained to the data set vital record specification can only contain movement information.
- Both data set vital record specification and name vital record specifications can contain movement information.

- Vital record specifications are chained using NEXT.
- Release options are not supported but can be defined.

What You Can Chain When You Specify VRSEL(NEW)

Setting the parmlib member EDGRMMxx OPTION VRSEL(NEW) operand, you can specify vital record specifications as follows:

- Both data set vital record specifications and name vital record specifications can contain retention information.
- The name vital record specification can use any retention type.
- Both data set vital record specification and name vital record specification can contain movement information.
- Vital record specification chains are made by using the NEXTVRS operand or the ANDVRS operand.
- Release options are fully supported.

Specifying Policies

To easily specify retention and movement policies, we recommend organizing policy values in three categories:

- Retention policy operands including COUNT, CYCLES, BYDAYSCYCLE, DAYS, EXTRADAYS, LASTREFERENCEDAYS, WHILECATALOG, and UNTILEXPIRED.
- Movement policy operands including DELAY, LOCATION, and STORENUMBER.
- Operands to manage the vital record specification itself including DELETEDATE and OWNER which are defaults in the RMM ADDVRS subcommand.

When you use the DFSMSrmm ISPF dialog to specify policies, the information in each panel is grouped as retention, movement, or vital record specification management fields. So you might find using the dialog easier than using the RMM TSO subcommands.

For example, Figure 85 shows two vital record specifications for retaining DFSMSHsm data sets.

Figure 86 on page 135 and Figure 87 on page 135 show the same policy values defined using the DFSMSrmm ISPF dialog panels.

```
RMM ADDVRS DSN('HSM.DMP.WEEKLYMA.**') NOGDG NEXTVRS(HSMVRS01) -
    DAYS COUNT(99999) DELAY( 0) -
    OWNER(LIBRARY ) DELETEDATE(1999/365) -
    LOCATION(06 ) STORENUM( 10) PRIORITY( 0)
RMM ADDVRS NAME(HSMVRS01) -
    DAYS COUNT(99999) DELAY( 0) -
    OWNER(LIBRARY ) DELETEDATE(1999/365) -
    LOCATION(HOME ) STORENUM(99999) PRIORITY( 0)
```

Figure 85. Specifying a Vital Record Specification With Default Operands

Panel Help	

EDGPV210	DFSMSrmm Add Data Set VRS
Data set mask : 'HSM.DMP.WEEKLYMA.**'	GDG . . NO
Job name mask :	
Count 99999	Retention type DAYS
	While cataloged no
Delay 0 Days	Until expired no
Location 06	
Number in location . 5	
Priority 0	
Next VRS in chain . . HSMVRS01	Release options:
Chain using . . . NEXT	Expiry date ignore yes
	Scratch immediate yes
Owner LIBRARY	
Description . . .	
Delete date . . . 1999/365	(YYYY/DDD)
Press ENTER to ADD the VRS, or END command to CANCEL.	
Command ==>	

Figure 86. Specifying a Vital Record Specification Using the Dialog Part 1

Figure 87 shows the same information as shown in the second vital record specification defined in Figure 85 on page 134.

Panel Help	

EDGPV240	DFSMSrmm Add Name VRS
Name : HSMVRS01	
Count 99999	Retention type DAYS
	While cataloged no
	Until expired no
Location HOME	
Number in location . 2	
Next VRS in chain . .	
Chain using . . .	
Owner LIBRARY	
Description . . .	
Delete date . . . 1999/365	(YYYY/DDD)
Press ENTER to ADD the VRS, or END command to CANCEL.	
Command ==>	

Figure 87. Specifying a Vital Record Specification Using the Dialog Part 2

Defining the Minimum Required Policy

You can set the EDGRMMxx parmlib member OPTION VRSMIN operand to control how many vital record specifications you want as the minimum for your installation. Also, you can control what RMM does when you do not have enough vital record specifications defined. Using the operand default values, if you have not defined the minimum of one vital record specification, RMM issues message edg2229I, and RMM inventory management stops.

You can define system-wide retention policies for all data sets not covered by other vital record specifications. If you have no other vital record specifications, define a single vital record specification with a data set name mask of '*' to establish a system-wide default. When no other vital record specifications match more specifically, then DFSMSrmm uses the '*' vital record specification to manage the data set. With VRSEL(NEW), you can have two vital record specifications match to a volume.

If you define a vital record specification with the '*' data set name mask and specify VRSEL(NEW), DFSMSrmm only uses one vital record specification to retain the data set. If you define a vital record specification with the '*.*' data set name mask and use VRSEL(NEW), DFSMSrmm treats the vital record specification with the '*.*' data set name mask as the primary vital record specification and looks for a secondary vital record specification. If you use vital record specification management values, only the '*' data set name mask can be used to specify system-wide retention values.

To prevent data sets from being retained too long, we recommend that the global policy be defined so that data sets are retained for a specified period, such as only as long as they are cataloged. The RMM ADDVRS subcommand you issue is:

```
RMM ADDVRS DSN('*') WHILECATALOG
```

To prevent data sets from being retained too long, you can force the retention to override any JCL-specified retention by using the RMM ADDVRS subcommand RELEASE(EXPIRYDATEIGNORE) operand. If you want to continue to allow JCL-specified retention dates, do not use RELEASE(EXPIRATIONDATEIGNORE).

Writing Your Own Vital Record Specifications

To write your own vital record specifications:

1. Review the examples in this section.
2. Tailor the examples by changing the data set names, location names, and retention information to match the values used in your location.

Example 1 — Retaining Catalogued Data Sets

Retain data sets that match to the data set name mask as long as they are cataloged but for a minimum of 5 days. If they are not cataloged, then retain them for a minimum of 5 days.

```
RMM ADDVRS DSN('WOODY.*') -           /* Data set VRS           */
          WHILECATALOG -               /* Retain while cataloged */
          LOCATION(HOME) -             /* Where to retain        */
          NEXTVRS(DAYS5) -             /* Name of next VRS       */
RMM ADDVRS NAME(DAYS5) -               /* Name VRS               */
```


DAYS	-	/* Retain by elapsed days	*/
COUNT(5)	-	/* Number of days	*/
LOCATION(HOME)		/* Where to retain	*/

Example 2 — Retaining Uncatalogued Data Sets

Retain data sets in storage location STORE1 that match the data set mask as long as they are cataloged. Then move the data sets to the home location and retain them for 5 days in the home location before making them eligible for release.

If the data sets are never cataloged, DFSMSrmm still retains them based on the parmlib CATRETPD operand. The CATRETPD operand specifies the number of hours that a data set should be retained before considering it not cataloged.

RMM ADDVRS DSN('MA.**')	-	/* Data set VRS	*/
WHILECATALOG	-	/* Retain while cataloged	*/
LOCATION(STORE1)	-	/* Where to retain	*/
NEXTVRS(XTRA5)	-	/* Name of next VRS	*/
RMM ADDVRS NAME(XTRA5)		/* Name VRS	*/
EXTRADAYS	-	/* Retain for extra days	*/
COUNT(5)	-	/* Number of days	*/
LOCATION(HOME)		/* Where to retain	*/

Note: When using the WHILECATALOG retention type, the CATRETPD value affects how long a data set might be retained. For example, if CATRETPD(9999) is specified, the data set is retained for 416 days even if it was never cataloged.

Example 3 — Retaining Cycles of Data Sets

Retain the latest 3 cycles of data sets that match to the data set name mask in the home location. A data set cycle is defined as one occurrence of a data set. Retain all additional cycles of the data sets that are not older than 3 days:

RMM ADDVRS DSN('GW.**')	-	/* Data set VRS	*/
LOCATION(HOME)	-	/* Where to retain	*/
CYCLES	-	/* Retain by cycles	*/
COUNT(3)	-	/* Number of cycles	*/
STORENUMBER(3)	-	/* Number of cycles to retain	*/
NEXTVRS(DAYS3)		/* Name of next VRS	*/
RMM ADDVRS NAME(DAYS3)	-	/* NAME VRS	*/
DAYS	-	/* Retain by elapsed days	*/
COUNT(3)	-	/* Number of days	*/
LOCATION(HOME)		/* Where to retain	*/

Example 4 — Moving Data Sets to Different Locations

Retain 3 cycles of the data set that matches the data set name mask. Retain each additional cycle of the data set for at least 3 days. Retain the latest cycle in the home location, the next cycle in storage location REMOTE, and the remaining cycles in the home location.

RMM ADDVRS DSN('WK.**')	-	/* Data set VRS	*/
CYCLES	-	/* Retain by cycles	*/
COUNT(1)	-	/* Number of cycles	*/
LOCATION(HOME)	-	/* Where to retain	*/
NEXTVRS(REMC1)		/* Name of next VRS	*/
RMM ADDVRS NAME(REMC1)	-	/* NAME VRS	*/
CYCLES	-	/* Retain by cycles	*/
COUNT(1)	-	/* Number of cycles	*/
LOCATION(REMOTE)	-	/* Where to retain	*/
NEXTVRS(HOMC1)		/* Name of next VRS	*/

```

RMM ADDVRS NAME(HOMC1) - /* NAME VRS */
CYCLES - /* Retain by cycles */
COUNT(1) - /* Number of cycles */
LOCATION(HOME) - /* Where to retain */
NEXTVRS(DAYS3) - /* Name of next VRS */
RMM ADDVRS NAME(DAYS3) - /* NAME VRS */
DAYS - /* Retain by elapsed days */
COUNT(3) - /* Number of days */
LOCATION(HOME) - /* Where to retain */

```

Example 5 — Retaining Data Sets Created within a Time Period

Retain data sets that match the data set name mask that have been created within the last 10 days. The data sets must also be cataloged and have been referenced within the last 2 days.

```

RMM ADDVRS DSN('TEST.AND') - /* Data set VRS */
LASTREFERENCEDAYS - /* Retain based on last referenced */
COUNT(2) - /* Number of days */
ANDVRS(DAYS10) - /* Chain using AND VRS */
RMM ADDVRS NAME(DAYS10) - /* NAME VRS */
DAYS - /* Retain by elapsed days */
COUNT(10) - /* Number of days */
ANDVRS(WC) - /* Chain using AND VRS */
RMM ADDVRS NAME(WC) - /* NAME VRS */
WHILECATALOG - /* Retain while cataloged */

```

Note: In the example, the ANDVRS operand is used which means that all the retention criteria must be met for the data set to be retained. Also the LOCATION(HOME) is not specified because it is the default and need not be specified.

Example 6 — Holding Data Sets for Extra Days

Retain each data set that matches the data set name mask in the home location for 3 days. Then move the data set to the storage location VAULT1 for one cycle. Finally move the data set to the storage location STOREX and keep the data sets in STOREX for 30 days. After the 30 days are over, continue to retain the data sets in STOREX as long as the data sets are cataloged. When the data sets are no longer cataloged, return them to the home location and keep them there for 2 days before making them eligible for release. Using the LOCATION(CURRENT) in the example allows you to chain the STEXWC and HOLD2 vital record specifications to many other policies without regard to which storage location is used.

```

RMM ADDVRS DSN('PROD.OFF.**') - /* Data set vrs */
DAYS - /* Retain by elapsed days */
COUNT(3) - /* Number of days */
STORENUMBER(3) - /* Retain in storage location */
NEXTVRS(V1C) - /* Name of next vrs */
RMM ADDVRS NAME(V1C) - /* NAME vrs */
CYCLES - /* Retain by cycles */
COUNT(1) - /* Number of cycles */
LOCATION(VAULT1) - /* Where to retain */
NEXTVRS(STEX) - /* Name of next vrs */
RMM ADDVRS NAME(STEX) - /* NAME vrs */
EXTRADAYS - /* Retain for extra days */
COUNT(30) - /* Number of days to retain */
LOCATION(STOREX) - /* Where to retain */
NEXTVRS(STEXWC) - /* Name of next vrs */
RMM ADDVRS NAME(STEXWC) - /* NAME vrs */
WHILECATALOG - /* Retain while cataloged */
LOCATION(CURRENT) - /* Where to retain */
NEXTVRS(HOLD2) - /* Name of next vrs */

```

```

RMM ADDVRS NAME(HOLD2) - /* NAME vrs */
                      - /* Retain for extra days */
                      COUNT(2) - /* Number of extra days */

```

Example 7 — Retaining Generation Data Group Data Sets

Retain all copies of data sets that match the generation data group base name for 1 day prior to moving one cycle off-site to the STORAGE LOCATION VLT1. Retain 29 cycles of the data sets in the home location.

```

RMM ADDVRS DSN('HYMILLER.TEST.GDG') - /* Data set vrs */
                      DAYS - /* Retain for elapsed days */
                      GDG - /* Data sets are GDGs */
                      LOCATION(HOME) - /* Where to retain */
                      COUNT(1) - /* Number to retain */
                      STORENUMBER(1) - /* Number to retain */
                      NEXTVRS(N1) - /* Name of next vrs */
RMM ADDVRS NAME(N1) - /* NAME vrs */
                      CYCLES - /* Retain by cycles */
                      LOCATION(VLT1) - /* Where to retain */
                      COUNT(30) - /* Number of cycles */
                      STORENUMBER(1) - /* Number to retain */

```

Note: In the example, the GDG operand tells DFSMSrmm to use the GDG data set base name and the Gnnnn.Vnnn. suffix for the data set name.

Example 8 — Retaining Recently Used Data Sets

Retain the two most recently used data sets that match the data set name mask. In this example, the first and the fourth data sets are retained.

The following data sets are created in the following order:

1. DSN=JACK last referenced 1 day ago
2. DSN=JACK last referenced 12 days ago
3. DSN=JACK last referenced 12 days ago
4. DSN=JACK last referenced 1 day ago

```

RMM ADDVRS DSNAME('JACK') - /* Data set vrs */
                      COUNT(2) - /* Number of cycles */
                      CYCLES - /* Retain by cycles */
                      ANDVRS(JIM) - /* Name of next vrs */
RMM ADDVRS NAME(JIM) - /* NAME vrs */
                      COUNT(2) - /* Number of days */
                      LASTREFERENCEDAYS - /* Retain since last referenced */

```

Note: In this example, the ANDVRS operand tells DFSMSrmm that all the retention policies must be met for the data sets to be retained.

Example 9 — Using Management Value to Retain Data Sets

Retain data sets for 50 days by assigning the keyword date 98050 to the data set. Use EDGUX100 to assign a management value of D98050 to the data set, based on the EXPDT=98050 JCL keyword.

Retain the two most recent 2 cycles of the data set and move them to the storage location STORE1. Retain older cycles in the home location for 50 days since creation.

```

RMM ADDVRS DSNAME('D98050')          - /* Data set vrs          */
      DAYS                          - /* Retain by elapsed days */
      COUNT(50)                     - /* Number of days to retain */
      LOCATION(HOME)                 /* Where to retain        */
RMM ADDVRS DSNAME('USER.OFFSITE.DATA.**') - /* Data set vrs          */
      CYCLES                         - /* Retain by cycles       */
      COUNT(2)                      - /* Number of cycles       */
      LOCATION(STORE1)               /* Where to retain        */

```

Note: DFSMSrmm uses the management value when VRSEL(NEW) is specified.

With VRSEL(OLD), the policy matching the management value is ignored.

Example 10 — Returning Volumes to Scratch Status

Retain data sets created by an application as long as they are cataloged. When created normally, the data sets have an expiration date based on the parmlib RETPD default retention and are cataloged.

Define a second retention policy for those data sets that are created by jobs that abnormally end. Retain the data sets for 2 days to allow the application data to be validated. When the data set is no longer retained by this vital record specification, the volume expiration date is to be ignored and the volume moved to pending release, unless some other data set on the volume is unexpired. Then the volume can be returned to scratch immediately during the next run of inventory management vital record processing.

```

RMM ADDVRS DSNAME('ABEND')          - /* Data set vrs          */
      DAYS                          - /* Retain by elapsed days */
      COUNT(2)                     - /* Number of days         */
      LOCATION(HOME)                 /* Where to retain        */
      RELEASE(EXPIRYDATEIGNORE SCRATCHIMMEDIATE) /* Release actions      */
RMM ADDVRS DSNAME('APPL1.**')        - /* Data set vrs          */
      WHILECATALOG                  /* Retain while cataloged */

```

Example 11 — Retaining Data Sets Using Expiration Date

Data sets with a high-level qualifier of 'A' are created and cataloged. Retain the data sets for a maximum of 10 days in the home location but only until the data set expires. The data set expiration date is specified in the JCL using EXPDT=nnnnn. Use the EDGUX100 exit to assign a management value of MVnnnnn to the data set. When data sets have reached their expiration date or after 10 days, the data sets are moved to the storage location MAINZ for 5 more days.

```

RMM ADDVRS DSN('A.**')              - /* Data set VRS          */
      UNTILEXPIRED                  - /* Retain until expired   */
      DAYS                          - /* Retain by elapsed days */
      COUNT(10)                     - /* Number of days         */
      LOCATION(MAINZ)                - /* Where to retain        */
      NEXTVRS(D2)                   - /* Name of the next VRS   */
RMM ADDVRS NAME(D2)                 - /* Name VRS               */
      EXTRADAYS                     - /* Retain for extra days  */
      COUNT(5)                      - /* Number of days         */
      LOCATION(MAINZ)                /* Where to retain        */
RMM ADDVRS DSN('MV*')              - /* Data set VRS MV        */
      WHILECATALOG                  - /* Retained while cataloged */
      LOCATION(WARWICK)             /* Where to retain        */

```

If a data set remains cataloged, it is retained 10 days in HOME location, then 5 days in storage location MAINZ and finally moves to the storage location

WARWICK where it remains until it is no longer cataloged. The data set returns to the home location when it is no longer cataloged.

If a data set is uncataloged before the 10 days specified in the vital record specification, then the data set is retained for example 7 days in HOME location. Then the data set move to the storage location MAINZ where the data sets are retained for 5 days. After 5 days, the data sets will return to the home location.

Example 12 — Combining Vital Record Specifications

Data sets with a high-level qualifier of 'A' are created and cataloged. Retain the data sets for 10 days in the home location and then in the storage location MAINZ until they expire. The data set expiration is specified through use of management values that correspond to vital record specifications, as in Example 11. MV* is a generic vital record specification that matches to a vital record management value that starts 'MV'. The vital record specification retains data sets in Warwick while cataloged, and then in Tucson 10 days after they are uncataloged. In the example UNTILEXPIRED is used, so the MV* vital record specification is never applied.

```
RMM ADDVRS DSN('A.**')      -          /* Data set VRS          */
              LOCATION(HOME) -          /* Location              */
              DAYS          -          /* Retain by elapsed days */
              COUNT(10)     -          /* Number of days        */
              NEXTVRS(D2)   -          /* Name of next VRS      */
RMM ADDVRS NAME(D2)        -          /* Name VRS              */
              UNTILEXPIRED  -          /* Retain until expired   */
              LOCATION(MAINZ) -        /* Where to retain       */
RMM ADDVRS DSN('MV*')      -          /* Data set VRS          */
              WHILECATALOG  -          /* Retain while cataloged */
              LOCATION(WARWICK) -      /* Where to retain       */
              NEXTVRS(M2)   -          /* Name of Next VRS      */
RMM ADDVRS NAME(M2)        -          /* Name VRS              */
              EXTRADAYS     -          /* Retain for extra days  */
              COUNT(10)     -          /* Number of days        */
              LOCATION(TUCSON) -        /* Where to retain       */
```

Note: In the example, the data sets do not move to the storage location TUCSON because the retention period for both the vital record specifications is the same.

For a data set that remains cataloged, the data set is retained for 10 days in the home location. Then the data set moves to the storage location MAINZ and remains there until the data set has been uncataloged for 10 days. After the 10 days, the data set returns to the home location.

For a data sets that is uncataloged after 7 days; it is retained 10 days in HOME location, then 7 days in storage location MAINZ location; 10 extra days since they were uncataloged.

```
c-----n-----r      data set name VRS
c-----u-----r      management value VRS
0       7   10   17
c=created, u=uncataloged, n=next vrs, r=released
```

Chapter 8. Storage Location Management for System-managed Tape Virtual Tape Servers

This chapter describes DFSMSrmm vaulting support for logical volumes that are exported from and imported into virtual tape servers (VTS).⁹

DFSMSrmm Export Processing

1. Define vital record specifications that specify locations to which data sets in the VTS should be moved.
2. Set the destination locations for logical volumes by running DFSMSrmm inventory management storage location management. DFSMSrmm sets the destination that is based on the required location that is set previously.
DFSMSrmm allocates a bin number to the volume if the volume's destination is shelf managed. There will be unused bin numbers for logical volumes that are moving to shelf-managed storage locations. You will waste bin numbers because EDGRPTD ignores the assigned bin numbers for logical volumes.
3. Create a list of volumes to export. Create the list after you run storage location management which sets the destination for the volume for any new required moves. The required moves are volume moves identified during vital record processing or by changes made when you issue the RMM CHANGEVOLUME subcommand. Use the RMM SEARCHVOLUME subcommand to obtain volume information for the list as shown in Figure 88. DFSMSrmm returns a list of volumes to export that you can use as input for creating the export list logical volume file 1. You can use the EDGJIMPC sample to reformat the CLIST file for use in the export list.

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) LOCATION(vts)-  
DESTINATION(dest) CLIST(',dest') -  
INTRANSIT(N)
```

Figure 88. Creating a Volume Export List

4. Start the export process as described in the *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.
The VTS export process runs asynchronously to DFSMSrmm processing. Once the export request has been initiated, VTS library signals trigger DFSMSrmm actions related to export processing.
During the VTS export process, logical volumes are copied to a stacked volume and the stacked volume is completed. For each logical volume copied to the stacked volume, DFSMSrmm is notified of both the logical volume and stacked volume. DFSMSrmm updates the DFSMSrmm control data set with the volume serial number of the stacked volume as the 'in container'.
5. Create movement report and pick lists using the DFSMSrmm EDGRPTD report utility.
When the export request ends, run DFSMSrmm inventory management report extract processing to obtain information about the logical volumes and stacked volumes. Use the DFSMSrmm EDGRPTD utility to create your movement reports and pick lists.

9. Import/Export support is available with APAR OW36342 or OW36343.

6. Transfer the stacked volumes in the export hold category to the exit station using the Library Manager console.
7. Physically move the stacked volumes listed in your movement report from the VTS exit station to the destination storage location.
8. Confirm that volumes have been moved. When the volumes have been moved, use the RMM CHANGEVOLUME subcommand as shown in Figure 89 to confirm the completion of the movement of volumes.

RMM CHANGEVOLUME * CONFIRMMOVE

Figure 89. Confirming Volume Moves for Exported Volumes

DFSMSrmm Volume Not In Library Support

DFSMSrmm provides volume not in library support. The DFSMSrmm EDGLCSUX installation exit returns information for a volume, indicating whether the volume is a logical, exported volume and the stacked volume on which it is exported. The sample CBRUXVNL exit is updated to call EDGLCSUX to issue a WTOR for all logical volumes. EDGLCSUX issues message EDG8123D to provide information about the stacked volume. Refer to *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information about DFSMSrmm volume not in library support.

DFSMSrmm Import Processing

DFSMSrmm supports the importing of any logical volumes, not just those defined to DFSMSrmm.

1. Use the Library Manager console to transfer stacked volumes to be imported from the Unassign category to the Import category.

During import processing, the imported volumes can be automatically added to the DFSMSrmm control data set if not currently defined or left for processing by another system. If the volume is already known to DFSMSrmm, the volume is accepted for processing if it is defined as an exported logical volume, otherwise the volume is rejected. DFSMSrmm does not add rack numbers for the imported volumes because rack numbers are not supported for logical volumes.
2. Importing one or more logical volumes can be initiated independently of DFSMSrmm. Create an import list volume and request the import using the CBRXLCS macro or the LIBRARY command.

To create the volume list for file 1 of the import list logical volume, you can search in the DFSMSrmm control data set, tailor the DFSMSrmm report extract file tailoring, or use any other method to identify the volumes to be imported.

You can use the RMM SEARCHVOLUME subcommand to build the list of logical volumes with their containing stacked volume and status as shown in Figure 90. Specify the TYPE(LOGICAL) operand and DFSMSrmm returns the container volume serial number, logical volume serial number, and volume status between your specified CLIST prefix and suffix strings. The resultant output file can be used as input for creating the import list logical volume file 1 after it is reformatted using the EDGJIMPC sample.

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) OWNER(*) DESTINATION(vts) -  
TYPE(LOGICAL) CLIST
```

Figure 90. Creating a Volume Import List

Completing the IMPORT Process

DFSMSrmm uses the cartridge entry processing installation exit to track logical volumes that are imported by the VTS. OAM calls the exit once for each logical volume imported. DFSMSrmm provides the same entry processing for both logical and physical volumes. For logical volumes, DFSMSrmm removes the stacked volume association.

Chapter 9. Requesting and Releasing Volumes

This chapter describes how to manually request a scratch volume and how to release a volume, automatically or manually.

DFSMSrmm assigns you a scratch volume automatically when you run a batch job that requests a non-specific tape mount. DFSMSrmm also automatically determines when that volume is eligible for release and schedules any release actions that have been specified for it. Typically you should not have to manually request or release volumes.

Requesting Scratch Volumes Manually

Related TSO Subcommand

Use the GETVOLUME subcommand to request a scratch volume. See “GETVOLUME: Requesting and Assigning Scratch Volumes” on page 326 for more information.

When you manually request a scratch volume without running a batch job, DFSMSrmm assigns you a volume with a status of user. This means the volume can be overwritten at any time by any user authorized to write to the volume.

To request a scratch volume:

1. Select Option A (REQUEST) on the Librarian Menu and press ENTER.
DFSMSrmm displays a Request a Volume panel such as the one shown in Figure 91.

Panel		Help	

EDGPT500		DFSMSrmm Request a Volume	
Command ==>			
Owner	MAZOWN44	Pool	
		or	
Owner Access		Media Name	
Retention Period . .		Expiration Date . . .	(YYYY/DDD)
Description			
Security Name . . .		Release Actions	
		Return to SCRATCH Pool . . YES	
		Replace volume NO	
MVS Use		Return to owner NO	
VM Use		Initialize volume NO	
		Erase Volume NO	
Location		Notify Owner YES	
Volume Access List:		Access	
User . .	User . .	User . .	User . .
User . .	User . .	User . .	User . .
User . .	User . .	User . .	User . .
Press ENTER to ADD the volume, or END command to CANCEL.			

Figure 91. DFSMSrmm Request a Volume Panel

2. All fields on the panel are optional. Press PF1 to see help panels with field-specific information.

You can specify any of the following:

- An owner ID other than your own, which is the default
- A retention period or an expiration date if you do not want to use the default retention period set up by your installation
- A pool ID if you want the volume to be chosen from a specific pool

3. Press ENTER.

DFSMSrmm selects a scratch volume from either a default scratch pool or from a pool you specified, and changes the status of the volume to user status.

To display the pool IDs defined for your location, type CONTROL VLPOOLS from the command or option line of any panel. In the TSO environment, use the LISTCONTROL VLPOOL subcommand.

To display the security classes defined for your location, type CONTROL SECURITY from the command or option line of any panel. In the TSO environment, use the LISTCONTROL SECCLS subcommand.

Releasing Volumes

Related TSO Subcommand

Use the ADDVOLUME subcommand to specify release actions for a volume when you initially add it to DFSMSrmm. Use the CHANGEVOLUME subcommand to specify release actions for a volume already defined to DFSMSrmm. See “ADDVOLUME: Adding Volume Information” on page 241 and “CHANGEVOLUME: Changing Volume Information” on page 282 for information on using the ADDVOLUME and CHANGEVOLUME subcommands. Use the ADDVRS subcommand to specify release actions using vital record specifications. See “ADDVRS: Adding a Vital Record Specification” on page 255 for information about the ADDVRS subcommand.

A master or user volume defined to DFSMSrmm is eligible for release when you run expiration processing and DFSMSrmm determines that:

- All data sets residing on the volume have expired.
- The expiration date set for the volume has been reached.
- Neither the volume nor any of the data sets on the volume are being retained by one or more vital record specifications.

Prior to returning a volume to scratch status, DFSMSrmm checks for *release actions*, which are actions you want to perform for the volume before it can be returned to scratch status. You can add or change the release actions when the volume is initially defined to DFSMSrmm or at any time before the volume is eligible for release. DFSMSrmm performs most of these actions automatically and waits for you to confirm those actions that must be done manually. When no release actions are specified or all release actions have been done and confirmed, a volume eligible for release is automatically returned to scratch status.

In addition to adding or changing volume information to define release actions, you can define policies to control when volumes are eligible for release described in “ADDVOLUME: Adding Volume Information” on page 241.

Setting Release Actions for a Volume

Table 11 describes the release actions you can specify for a volume:

Table 11. Release Actions for a Volume

Release Action	Type of Processing
Any one of the following:	
• Returning volumes to scratch status	Automatic when EDGINERS is scheduled
• Replacing volumes	Manual
• Returning volumes to their owners	Manual
Any one of the following:	
• Initializing and erasing volumes	Automatic when EDGINERS is scheduled
• Notifying owners when their volumes have expired or are being released	Automatic if NOTIFY(Y) is defined in parmlib and owner user ID and node are defined

To specify release actions for a volume:

- Use the Add Volume panel as shown in Figure 92 on page 150, or the ADDVOLUME subcommand, to specify release actions for the volume when you initially define the volume to DFSMSrmm.
- Use the Change Volume Details panel, or the CHANGEVOLUME subcommand to change or set release actions for a volume already defined to DFSMSrmm. You can specify release actions anytime before the volume is released.

If you use the DFSMSrmm ISPF dialog panels, such as the Add Volume panel shown in Figure 92 on page 150, DFSMSrmm displays the release actions you can specify for a volume. Enter YES next to the release actions of your choice. Specify NO if you do not want a release action performed.

Panel Help Scroll		
EDGPT210		DFSMSrmm Add Volume
Command ==>		Page 1 of 2
Volume	N00001	Pool
Count		or
Media name	CART	Rack
Storage group		Location name
Retention Period	90	Expiration Date (YYYY/DDD)
Status	USER	Initialize
Description		
Data Set Name		
Media type	CST	Release Actions
Label	SL	Return to SCRATCH pool YES
Density		Replace volume NO
Recording format		Return to owner NO
Compaction		Initialize volume NO
Attributes		Erase volume NO
		Notify owner NO
Press ENTER for additional volume information, or END command to CANCEL.		

Figure 92. Release Actions on the DFSMSrmm Add Volume Panel

Returning Volumes to Scratch Status—Automatic Action

When you specify

Return to SCRATCH pool ==> YES

DFSMSrmm automatically returns the volume to scratch status, once any other release actions specified for the volume are completed.

Return to scratch is mutually exclusive with Replace volume and Return to owner.

When a volume returns to scratch status, its volume access is reset to NONE, and the following information is cleared so the volume can be reused: volume description, job name, accounting information, access list and owner access. Additionally, if the volume was associated with a software product defined to DFSMSrmm, the product number, feature code and level are all cleared, and the volume is removed from the list of volumes associated with the product.

Replacing Volumes—Manual Action

When you specify

Replace volume ==> YES

you set the replace volume release action for a volume. DFSMSrmm identifies the volume as needing to be replaced when the volume becomes eligible for release.

You replace volumes when they have permanent I/O errors, or when they reach the end of their useful life. DFSMSrmm tracks the number of I/O errors recorded for a volume and identifies it as needing to be replaced as long as you have not manually released the volume and are running DFSMSrmm expiration processing. Tape librarians must manually replace these volumes and confirm to DFSMSrmm that the action has taken place.

You can request that DFSMSrmm create a list of volumes to be replaced, using one of the following:

- The Volume Action Status panel.

Specify S against the REPLACE action in the list of actions.

or

- The SEARCHVOLUME subcommand, as follows:

```
RMM SEARCHVOLUME VOLUME(volser) ACTION(REPLACE) LIMIT(*) OWNER(*)
```

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information on the SEARCHVOLUME subcommand.

Once you have replaced a volume, you must confirm the action to DFSMSrmm. This ensures that the control data set is updated with the most current information, and allows other release actions, such as returning the volume to scratch status, to be processed. When the replace action is confirmed, DFSMSrmm automatically sets the initialize action for the volume so that the new volume can be labeled correctly.

Returning Volumes to Their Owners—Manual Action

When you specify

```
Return to owner ==== YES
```

DFSMSrmm marks the volume as needing to be returned to its owner. Tape librarians must manually return such volumes and confirm when the action has taken place.

You might want to use the owner address in the DFSMSrmm control data set to help you ensure volumes return to their owners.

After returning the volume, you must confirm to DFSMSrmm that the action has been completed. This ensures that the DFSMSrmm control data set is updated with the most current information. When you confirm that a volume has been returned to its owner, the volume and its related information is deleted from the control data set, and its rack number is left empty for reassignment. See “Confirming Manual Release Actions to DFSMSrmm” on page 155 for more information on confirming actions to DFSMSrmm.

To create a list of volumes waiting to be returned to their owners, use either:

- The Volume Action Status panel.

Specify S against the RETURN action in the list of actions.

or

- The SEARCHVOLUME subcommand, as follows:

```
RMM SEARCHVOLUME VOLUME(volser) ACTION(RETURN) LIMIT(*) OWNER(*)
```

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information on the SEARCHVOLUME subcommand.

Initializing and Erasing Volumes—Automatic Action

When you specify

Initialize volume ===> YES

and

Erase volume ===> YES

DFSMSrmm automatically erases and initializes a volume upon release. You can also specify either release action separately. Because these actions require that volumes be mounted, they should be specified only for exceptions, such as high security volumes.

DFSMSrmm automatically performs initialization and erase actions through the EDGINERS utility. We recommend that you include EDGINERS in your regularly scheduled inventory management activities. After EDGINERS erases and initializes volumes, it automatically updates the control data set to show that the actions have been completed. When EDGINERS is used to erase a volume, the volume is also initialized, so both actions can be performed with a single mount request. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on scheduling inventory management and on using EDGINERS.

Relabeling a Volume

When relabeling a volume defined to DFSMSrmm, DFSMSrmm uses the information from the old volume to define the new volume in the control data set.

The following information is taken from the old volume record: Expiration Date, Density, Use Count, Store Id, Bin Number, Old Bin Number, Loan Location, Previous Location, Last Read and Write dates, Assigned Date and Time, Owner, Status, Label Type, Release actions, Actions Pending (excluding initialize and erase), Volume access and accessors, Unit name, Rack number, Temporary / Permanent Read / Write error counts.

If you want to remove all the old information from the volume record, use the RMM DELETEVOLUME subcommand to remove the volume record and then issue the RMM ADDVOLUME subcommand to add volume information. If you are authorized to the STGADMIN.EDG.MASTER and STGADMIN.EDG.FORCE security resources, you can use the RMM CHANGEVOLUME FORCE operand to remove old information.

If you have an alternative means of erasing volumes, such as a magnetic degaussing machine for bulk erasure, or if you use an alternative utility to initialize volumes, you must confirm to DFSMSrmm when you have completed the erase or initialize actions to update the DFSMSrmm control data set and to free the processing of other release actions specified for the volume. When you confirm that a manual erase action was completed, DFSMSrmm automatically sets the initialize action for the volume because degaussing the volume destroyed the volume label.

Note: Do not degauss 3590 media as the servo tracks will be destroyed.

To create a list of volumes needing erasing or initialization, use one of the following:

- The DFSMSrmm Volume Action Status panel
Specify S against the INIT or ERASE actions in the list of release actions.
or

- The SEARCHVOLUME subcommand, as follows:

```
RMM SEARCHVOLUME VOLUME(volser) ACTION(INITIALIZE) LIMIT(*) OWNER(*)
```

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information on the SEARCHVOLUME subcommand.

Notifying Owners—Automatic Action

When you specify

```
Notify owner          ==> YES
```

DFSMSrmm automatically builds a message and sends it to the volume’s owner once DFSMSrmm marks the volume as pending release. In addition to setting the release action, the DFSMSrmm parmlib OPTION NOTIFY operand must be set to Y, and a valid user ID and node for the owner must be defined to DFSMSrmm.

For information on the message content and how to modify it, and for information on setting parmlib options, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

You can also specify RELEASEACTION(NOTIFY) as part of the GETVOLUME, ADDVOLUME, or CHANGEVOLUME subcommands. See “GETVOLUME: Requesting and Assigning Scratch Volumes” on page 326, “ADDVOLUME: Adding Volume Information” on page 241, and “CHANGEVOLUME: Changing Volume Information” on page 282 for more information.

To ensure that DFSMSrmm can send the message to the owner of the volume being released, you must specify both a user ID and node when you add an owner to DFSMSrmm. Use either the Add Owner Details Userid panel or the ADDOWNER subcommand to do this. Use the Change Owner Details Userid panel or the CHANGEOWNER subcommand to add a valid electronic address if the owner is already defined to DFSMSrmm.

See “Changing Owner Information” on page 64, “ADDOWNER: Adding Owner Information” on page 233, or “CHANGEOWNER: Changing Owner Information” on page 277 for more information.

If you specify NOTIFY OWNER for an owner that has no electronic mail address, DFSMSrmm cannot perform automatic notification. Additionally, DFSMSrmm cannot release the volume until you confirm that you have notified the owner.

To create a list of owners to be notified, use one of the following:

- The Volume Action Status panel
Type S against the NOTIFY action in the list of release actions.
or

- The SEARCHVOLUME subcommand, as follows:
RMM SEARCHVOLUME VOLUME(*volser*) ACTION(NOTIFY) LIMIT(*) OWNER(*)

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information on the SEARCHVOLUME subcommand.

You can use the report file extract to build a consolidated list of volumes and send a single message per owner, rather than the message per volume that DFSMSrmm sends.

Changing a Volume's Release Date

You can change or override the date a volume is to be considered for release in the following ways:

- Release a volume manually before it becomes eligible for release.
- Define a vital record specification to retain the volume or a data set on the volume.
- Change the expiration date or the retention period set for the volume.

Releasing a Volume Early

You can release a volume early, before all data sets on the volume have expired, and before the end of any retention period set for the volume or data sets on the volume by one or more vital record specifications.

If an expiration date or retention period was coded in the JCL when the data was originally written to the volume, the tape label is expiration date protected, and DFSMSrmm records this as the original expiration date for the volume. Normally, this date will have passed when a volume is released. The original expiration date never changes when you change the expiration date or retention period for the volume. Your installation should consider how to deal with those instances when a volume is released early.

See “Releasing Volumes Manually” on page 158 for details on manually releasing volumes. You can also use the DELETEVOLUME subcommand with the RELEASE operand; see “DELETEVOLUME: Deleting Volume Information” on page 320 for more information.

Defining Vital Record Specifications to Retain the Volume

You can define one or more vital record specifications for data sets on the volume, or for the volume itself, to set a new retention period. Use the ADDVRS subcommand or the DFSMSrmm ISPF dialog to define vital record specification. See “ADDVRS: Adding a Vital Record Specification” on page 255 or “Chapter 6. Retaining and Moving Your Volumes” on page 99 for more information.

Changing the Expiration Date for a Volume

The expiration date recorded for a volume is determined when each data set is written to the volume. You can change this date to extend the time before a volume is released, without writing to the volume again.

Use the CHANGEVOLUME subcommand with the EXPDT or RETPD operands, or the Change Volume panel to change the expiration date or the retention period for a volume. You can change the expiration date or the retention period anytime after the volume has been defined to DFSMSrmm and before the expiration date is reached. Retention periods set by one or more vital record specifications defined for the volume or for one or more data sets on the volume might override this expiration date.

See “CHANGEVOLUME: Changing Volume Information” on page 282 or “Changing Volume Information” on page 60 for more information on changing the expiration date.

Confirming Manual Release Actions to DFSMSrmm

DFSMSrmm always confirms the release actions it performs automatically. These are:

- Returning volumes to scratch status
- Erasing and initializing volumes
- Notifying owners

When automatic processing is not available or if you do not use it, you must perform release actions manually and then confirm to DFSMSrmm that those actions have been completed.

Manual actions are actions that are external to DFSMSrmm, such as using a degaussing device to erase tape volumes. By confirming manual actions, you enable DFSMSrmm to keep accurate records and you allow release processing for the volume to continue. DFSMSrmm cannot release a volume until you confirm that you have manually performed any pending release actions. You must confirm the replace and return release actions before confirming any others.

Use the Volume Action Status panel, or the RMM TSO CHANGEVOLUME subcommand to confirm manual release actions to DFSMSrmm. You can also use the Volume Action Status panel or the RMM TSO SEARCHVOLUME subcommand to create a list of volumes requiring a release action to be taken and confirmed, and use line operators from the list to confirm those actions.

Using the Volume Action Status Panel

To manually confirm one or more release actions:

1. Select Option 8 (CONFIRM) on the Volume Menu.

Note to Tape Librarians: You can bypass the Volume Menu by selecting Option 9 (CONFIRM) on the Librarian Menu.

2. Press ENTER. DFSMSrmm displays the Volume Action Status panel as shown in Figure 93 on page 156.
3. Type C against the type of release action you want to confirm. You can confirm as many types of release actions as you want.

Panel Help

EDGPC700 DFSMSrmm Volume Action Status ROW 1 TO 13 OF 13

COMMAND ==> SCROLL ==> PAGE

The following line commands are valid: C, U and S

Dest-

S Action Location ination Move Type Status

ERASE Unknown

INIT Pending

NOTIFY Pending

REPLACE Unknown

RETURN Unknown

SCRATCH Pending

MOVE LIBRARY DISTANT NOTRTS Pending

MOVE LOCAL LIBRARY NOTRTS Pending

C MOVE LOCAL SHELF RTS Pending

C MOVE LOCAL SHELF NOTRTS Pending

C MOVE REMOTE LIBRARY NOTRTS Pending

C MOVE REMOTE SHELF RTS Pending

C MOVE REMOTE SHELF NOTRTS Pending

***** BOTTOM OF DATA *****

Figure 93. EDGPC700 - DFSMSrmm Volume Action Status Panel

4. Press ENTER.

DFSMSrmm confirms all applicable actions.

You can also use the Volume Action Status Panel to request a list of volumes requiring release actions to be performed, and confirm those release actions from this list. To do this:

1. Type S against the release action for which you want to see a list of volumes. For example, to see a list of all volumes to be replaced, specify:

		Dest-			
S	Action	Location	ination	Move Type	Status
	REPLACE				Pending

2. Press ENTER.

DFSMSrmm returns a list such as the one in Figure 94 on page 157:

```

Panel  Help
-----
EDGPT610          DFSMSrmm Volume Action Summary List          Row 1 to 3 of 3
Command ==>          Scroll ==> PAGE

The following line commands are valid: C, CE, CI, CM, CN, CO, CR, E, L and V
Volume      Assigned   Expiration   Rack      Dest-   Tra-
S Serial Owner   Date       Date       Location Number Action ination nsit
-----
A04001 D021906 1996/106   1996/106   SHELF    A04001 S              N
A04010 D021906 1996/106   1996/106   SHELF    A04010 S              N
C A04012 D021906 1996/103   1996/103   SHELF    A04012 SI             N
***** Bottom of data *****

```

Figure 94. DFSMSrmm Volume Action Summary List

- Table 12 describes the line operators you can use against specific entries in the list to confirm pending release actions:

Table 12. Confirm List Line Operators

Line Operator	Description
C	Confirm any release action
CE	Confirm that the volume has been erased
CI	Confirm that the volume has been initialized
CM	Confirm that the volume has been moved.
CN	Confirm that the volume's owner has been notified
CO	Confirm that the volume has been returned to its owner
CR	Confirm that the volume has been replaced
L	List a multi-volume chain

Confirming a Single Release Action Using the RMM CHANGEVOLUME Subcommand

Use the CHANGEVOLUME subcommand with a specific volume serial number and the CONFIRMRELEASE operand to confirm a single release action. For example, to confirm the INIT release action for volume VOL001, enter:

```
RMM CHANGEVOLUME VOL001 CONFIRMRELEASE(INIT)
```

Use the CHANGEVOLUME subcommand with an asterisk and the CONFIRMRELEASE operand to confirm release actions for multiple volumes. When you specify an asterisk, DFSMSrmm confirms the action on the CONFIRMRELEASE operand for all volumes pending this release action.

For example, to confirm that you have replaced volumes for all volumes requiring this action, enter:

```
RMM CHANGEVOLUME * CONFIRMRELEASE(REPLACE)
```

Use SEARCHVOLUME with the CLIST operand to create a data set of executable CHANGEVOLUME subcommands to confirm a list of release actions. DFSMSrmm builds a CLIST data set for you. You can edit the CLIST to remove any volumes that did not get replaced and run the CLIST at your convenience.

Figure 95 shows how to build a list of CHANGEVOLUME subcommands to confirm all replace actions for only those volumes requiring a replace action.

```
RMM SEARCHVOLUME VOL(*) OWNER(*) LIMIT(*) -  
  CLIST('RMM CHANGEVOLUME ',' CONFIRMRELEASE(REPLACE)') -  
  RELEASEACTION(REPLACE)
```

Figure 95. Confirming Replace Actions for Volumes

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for information on using SEARCHVOLUME with the CLIST operand.

Releasing Volumes Manually

Related TSO Subcommand

Use the DELETEVOLUME subcommand to manually release one or more volumes. See “DELETEVOLUME: Deleting Volume Information” on page 320 for more information.

You can release a volume any time before all data sets on the volume have expired, before the expiration date set for the volume is reached, or before the end of the retention period set for the volume or data sets on the volume by one or more vital record specifications.

If you do not release a volume manually, DFSMSrmm automatically determines when it is eligible for release and schedules the release actions specified for it. If you release a volume while it resides in a storage location or while it is in transit between the removable media library and a storage location, or between storage locations, DFSMSrmm indicates that the volume is pending release, and processes any release actions specified for the volume when the volume returns to the removable media library. See “Setting Release Actions for a Volume” on page 149 for more information on release actions.

Tape librarians can release any volume or group of volumes defined to DFSMSrmm, regardless of ownership.

General users are only allowed to release volumes they own, and should use the Release Volumes panel available to them from the User Menu. See “Releasing a Volume Manually” on page 29 for more information.

To release one or more volumes:

1. Select option 4 (RELEASE) on the Volume Menu and press ENTER.

Note: Tape librarians can bypass the Volume Menu by selecting Option 8 (RELEASE) on the Librarian Menu.

DFSMSrmm displays the Release Volumes panel, as Figure 96 shows.

Panel	Help
EDGPT400	DFSMSrmm Release Volumes
Command ==>	
Volume	May be generic. Leave blank for all volumes
Enter optional parameters to qualify search (leave blank for specific volume):	
Owner	Enter a specific user, default is your userid
Status	SCRATCH, MASTER or USER.
Release type . .	RELEASE, REMOVE or FORCE. Used only if 1 volume is listed
Limit	Limit search to first nnnn volumes
The following line commands will be available if a list is displayed:	
F - Force volume	E - Eject volume
O - Display owner information	L - List multi-volume chain
R - Release USER or MASTER volume or remove SCRATCH volume	
V - Display volume information	
Press ENTER to SEARCH, or END command to CANCEL.	

Figure 96. DFSMSrmm Release Volumes Panel

2. Specify a volume serial number or leave blank to have DFSMSrmm create a list of volumes.
All other fields on the panel are optional. Press PF1 to see help panels with field-specific help.
3. Press ENTER. DFSMSrmm releases the volume unless you have specified the confirm option for your session or you requested a list of volumes.
 - If you requested the confirm option for your session, DFSMSrmm displays the Confirm Volume Release panel. This panel displays information about the single volume you are releasing and asks you to confirm that you want to release the volume. For example, DFSMSrmm displays a panel such as the one in Figure 97 on page 160:

Panel Help	

EDGPT410	DFSMSrmm Confirm Volume Release
Command ==>	
Volume : VOL000	Location : LIB1
Volume type. . . . : LOGICAL	In container . . . :
Media name : 3480	Rack Number . . . : VOL000
	Expiration Date . . : 1993/311
Status : USER	Original
	Expiration Date . . :
Description :	
Data Set Name . . . : DATA.SET.ONE	
Media type : CST	Release Actions
Label : SL	Return to SCRATCH pool . . : YES
Density : *	Replace volume : NO
Recording format . . : 18TRACK	Return to owner : NO
Compaction : IDRC	Initialize volume : NO
Attributes : NONE	Erase volume : NO
Availability . . . :	Notify owner : NO
Press ENTER to RELEASE Volume, or END command to CANCEL.	

Figure 97. DFSMSrmm Confirm Volume Release Panel

- If you requested confirm release for your session and you specified the E line operator from the list for a volume residing in either an automated or a manual tape library dataserver, DFSMSrmm displays the Confirm Volume Eject panel.

This panel displays information about the single volume you are ejecting and asks you to confirm that you want to eject the volume. For example, DFSMSrmm displays a panel such as the one in Figure 98:

Panel Help	

EDGPT430	DFSMSrmm Confirm Volume Eject
Command ==>	
Volume : VOL000	Location : LIB1
Media name : 3480	Rack Number . . . : VOL000
	Expiration Date . . : 1993/311
Status : USER	Original
	Expiration Date . . :
Description :	
Data Set Name . . . : DATA.SET.ONE	
Media type : CST	Release Actions
Label : SL	Return to SCRATCH pool . . : YES
Density : *	Replace volume : NO
Recording format . . : 18TRACK	Return to owner : NO
Compaction : IDRC	Initialize volume : NO
Attributes : NONE	Erase volume : NO
Availability . . . :	Notify owner : NO
Press ENTER to EJECT Volume, or END command to CANCEL.	

Figure 98. DFSMSrmm Confirm Volume Eject Panel

4. Press ENTER to confirm that the volume is to be ejected.

Note: If release actions are not correct, you can use the fast path command VOLUME CHANGE to display the Change Volume panel, make your changes, then continue with the release process. See “Changing Volume Information” on page 60 for more information.

5. If you requested that DFSMSrmm build a list of volumes, DFSMSrmm displays a volume list such as the list in Figure 99:

Panel Help Scroll									

EDGPT020		DFSMSrmm Volumes (Page 1 of 2)					Row 1 to 8 of 8		
Command ==>							Scroll ==> PAGE		
The following line commands are valid: C,I,O,R,E,L and V									
Use the RIGHT command to view other data columns									
S	Volume serial	Owner	Assigned date	Expiration date	Status	Location	Dest-ination	Tra-nsit	Data sets

	A05000	RMMUSER	1999/012	1999/017	MASTER	SHELF		N	0
	A05001	RMMUSER	1999/012	1999/017	MASTER	SHELF		N	0
	A05002	RMMUSER	1999/012	1999/017	MASTER	SHELF		N	0
	A05003	RMMUSER	1999/012	1999/017	MASTER	SHELF		N	0
	A05004	RMMUSER	1999/012	1999/017	MASTER	SHELF		N	0
	A05005	RMMUSER	1999/012	1999/017	USER	SHELF		N	0
	A05006	RMMUSER	1999/012	1999/017	USER	SHELF		N	0
	A05007	RMMUSER	1999/012	1999/017	USER	SHELF		N	0
***** Bottom of data *****									

Figure 99. DFSMSrmm List of Volumes

Table 13 describes the line operators that can be used to release specific volume entries in the list. The general user can use the R and V line operators. The librarian can use all the line operators shown in Table 13.

Table 13. DFSMSrmm Release Volumes Line Operators

Line Operator	Description
E	Eject volume
F	Remove the volume from the removable media library, delete from DFSMSrmm, and uncatalog all data sets on the volume for which DFSMSrmm has recorded information in the control data set
L	List a multi-volume chain
O	Display owner information
R	Release volumes based on specified release actions
V	Display full volume details

Chapter 10. Requesting Information about Your Resources

All DFSMSrmm users can request information about resources defined to DFSMSrmm, such as shelf locations, volumes, software products, data sets, and vital record specifications. Users can also request to view information about parmlib options and DFSMSrmm control data set control information.

You can use the DFSMSrmm ISPF dialog or the RMM TSO subcommands to obtain information about your resources. Information about obtaining information using the DFSMSrmm ISPF dialog or the RMM TSO subcommands begins in “Displaying Details about a Single Resource”.

Using DFSMSrmm Reporting Utilities and Sample Reports

DFSMSrmm also provides the DFSMSrmm utility EDGRPTD and sample reports you can use to obtain information about your resources. EDGRPTD uses the report extract data set produced by the EDGHSKP utility during inventory management to create different types of reports. By scheduling EDGRPTD to use the extract data set created after DFSMSrmm completes vital record specifications processing, you can have a drop/ship list that reflects the inventory management processing just performed.

One of the reports you can create using EDGRPTD is the Ready to Scratch Volumes Movement report. DFSMSrmm does not allow volumes to return to scratch until all pending movement and actions for the volume have been completed. All volumes must be returned to their home location before they can be returned to scratch. Ready to scratch volumes are volumes that have expired before returning to their home location and whose only pending action is the return to scratch release action. Using the Ready To Scratch report, you can separate volumes that are private or have release actions pending from those awaiting return to scratch.

See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more details on EDGRPTD and for information on scheduling inventory management utilities and reports DFSMSrmm provides for use with the extract data set.

Displaying Details about a Single Resource

You can request detailed information about any single resource defined to DFSMSrmm, such as a shelf location, volume, owner, software product, data set, or vital record specification, by using the display function in the dialog or any of the RMM TSO LIST subcommands.

Table 14 describes what to specify on the different display panels to request information about the resources defined to DFSMSrmm.

Table 14. Information to Specify on Display Panels

On The Display Panel for:	Specify:
Shelf location	A specific rack or bin number
	If you specify the bin number of a shelf location in a storage location, you must also specify the location.

Table 14. Information to Specify on Display Panels (continued)

On The Display Panel for:	Specify:
Volume	<p>A specific volume serial number</p> <p>You can also limit the amount of information DFSMSrmm displays about the volume by specifying which of four panels you want to view.</p>
Software product	<p>A specific software product number</p> <p>You can also specify a version of the software product.</p>
Data set	<p>A specific data set name and volume serial number</p> <p>If the data set for which you want to view information is not the first data set on the volume, specify a data set sequence number.</p>
Vital Record Specification	<p>Either a data set name or a volume serial number</p> <p>You can use a data set name mask or a generic volume serial number to request information about a vital record specification.</p>

To request detailed information about any single resource defined to DFSMSrmm:

1. Request a display panel, by doing one of the following:
 - Select option 1 (DISPLAY) on a resource menu and press ENTER.
For example, type 1 on the command line of the DFSMSrmm Volume Menu to request the Volume Display panel.>
or
 - Specify a fast path command and press ENTER.

For example, if you specify VOLUME DISPLAY from the command or option line of any DFSMSrmm panel, DFSMSrmm displays the Volume Display panel as shown in Figure 100 on page 165.

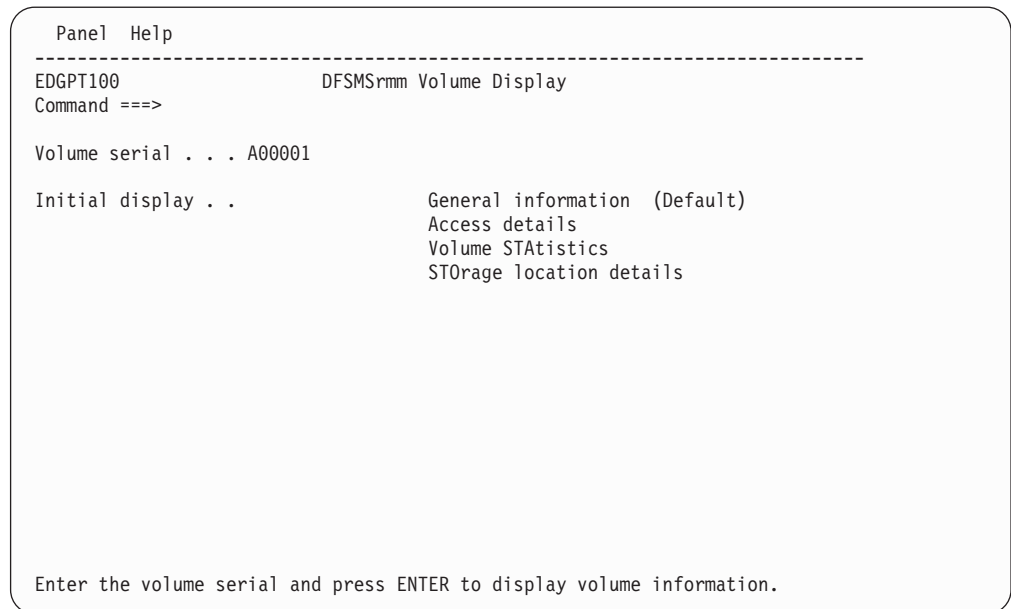


Figure 100. DFSMSrmm Volume Display Panel

2. Specify the unique number or name of the resource for which you want information. This number or name must be already defined to DFSMSrmm. For example, enter a volume serial number A00001 to request information about volume A00001, as shown in Figure 100.
3. Press ENTER.
DFSMSrmm displays a panel containing information about the volume you specified.

The following DFSMSrmm ISPF dialog panels show the kind of details DFSMSrmm displays. See the DFSMSrmm ISPF dialog help for descriptions of the fields.

Shelf Location Details

DFSMSrmm displays a Rack or Bin details panel, such as the one in Figure 101:

```
Panel  Help
-----
EDGPR110          DFSMSrmm Rack or Bin Details
Command ==>

Rack or Bin number . : A00001

Location . . . . . : SHELF
Media name . . . . . : 3480
Volume . . . . . : A00001
Status . . . . . : IN USE
```

Figure 101. DFSMSrmm Rack or Bin Details Panel

Volume Details

Figure 102 displays volume information, including: general volume information, volume access details, volume statistics, and storage location information.

```
Panel  Help
-----
EDGPT110          DFSMSrmm Volume Details - A05000
Command ==>

Volume . . . . . : A05000      Rack number . . . . . :
Media name . . . . . : 3480      Status . . . . . : MASTER
                                   More:      +
Volume type . . . . : PHYSICAL   Expiration date . . . . : 1999/017
Retention date . . . :           Original expiration date . . :
Description . . . . :

Data set name . . . :
Media type . . . . . : *
Label . . . . . : SL
Current version :
Required version :

Release actions:
Return to SCRATCH pool . : YES
Replace volume . . . . . : NO
Return to owner . . . . . : NO
```

Figure 102. DFSMSrmm Volume Details Panel

```

Density . . . . . : *
Recording format . . : *
Compaction . . . . . : *
Attributes . . . . . : NONE
Availability . . . . . :
Initialize volume . . . : NO
Erase volume . . . . . : NO
Notify owner . . . . . : NO
Expiry date ignore . . . : NO
Scratch immediate . . . : NO

Owner . . . . . : RMMUSER
Assigned date . . . : 1999/012
Owner access . . . . . : ALTER
Assigned time . . . . . : 04:44:44

Security name . . . :
Classification . . . :
Account number . . . :

Jobname . . . . . :
MVS use . . . . . : YES
VM use . . . . . : NO

Loan location . . . :
Last changed by . . . . : RMMUSER

Previous volume . . . :
Next volume . . . . . :

Volume access list :
  User . . . . . :
  User . . . . . :
  User . . . . . :
  User . . . . . :
Access . . . . . : NONE
  User . . . . . :
  User . . . . . :
  User . . . . . :
  User . . . . . :

Volume use count . . : 0
Create date . . . . : 1999/012
Date last read . . . :
Volume usage (Kb) . . . : 0
Create time . . . . . : 04:44:44
Date last written . . . :

Drive last used . . . :

Volume sequence . . : 1
Number of data sets . . . : 0
Data set recording . . . : ON

Errors:
  Temporary read . . : 0
  Permanent read . . : 0
  Temporary write . . . : 0
  Permanent write . . . : 0

Actions pending:
  Return to SCRATCH pool . . : NO
  Replace volume . . . . . : NO
  Return to owner . . . . . : NO
  Initialize volume . . . . . : NO
  Erase volume . . . . . : NO
  Notify owner . . . . . : NO

Location . . . . . : SHELF
Destination . . . . . :
Location type . . . :
In transit . . . . . : NO

In container . . . . : CCC

Storage group . . . :
Home location . . . . . : SHELF
Required location . . . . :
Move mode . . . . . : AUTO
Movement tracking date . . . :

Bin number . . . . . :
Media name . . . . . :
Old bin number . . . :
Media name . . . . . :

Product details:
  Product number . . . :
  Level . . . . . :
  Feature code . . . :

Enter SCROLL commands for more volume information, or END command to CANCEL.

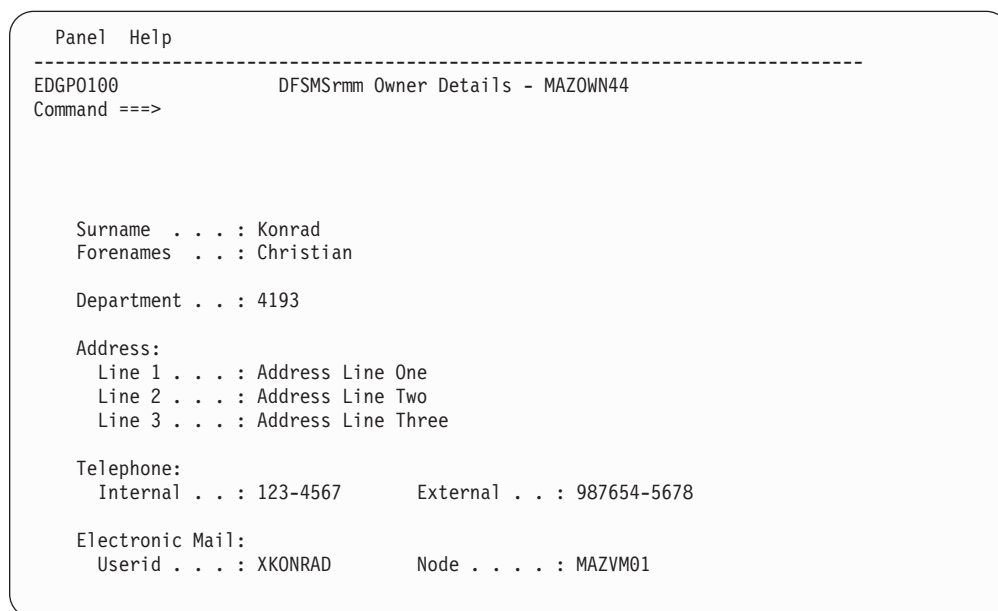
```

Figure 103. DFSMSrmm Volume Details Panel (Cont.)

Owner Details

To request information about an owner defined to DFSMSrmm, you must specify an owner ID on the Owner menu. DFSMSrmm displays an Owner Details panel containing detailed information about the owner ID you specified as shown in Figure 104.

If you use the OWNER DISPLAY fast path command to request this information, you must specify a specific owner ID as part of the command. For example, specify OWNER DISPLAY MAZOWN44 to view details about the owner defined by the owner ID MAZOWN44.



```
Panel  Help
-----
EDGP0100          DFSMSrmm Owner Details - MAZOWN44
Command ==>

Surname . . . : Konrad
Forenames . . : Christian

Department . . : 4193

Address:
Line 1 . . . : Address Line One
Line 2 . . . : Address Line Two
Line 3 . . . : Address Line Three

Telephone:
Internal . . : 123-4567      External . . : 987654-5678

Electronic Mail:
Userid . . . : XKONRAD      Node . . . . : MAZVM01
```

Figure 104. DFSMSrmm Owner Details Panel

Software Product Details

DFSMSrmm displays a Product Details panel, such as the one shown in Figure 105 on page 169.


```

Panel  Help
-----
EDGPP110                      DFSMSrmm Product Details          Row 1 to 2 of 2
Command ==>

Product Number . . . : PRODX01

Level . . . . . : V01R03M00          Owner . . . : MAZOWN22

Product Name . . . : TEST

Description . . . . : TEST

Volume details:
  Volume  Rack  Feature
  Serial  Number Code
  -----
  VOL490  RAC490 1234
  VOL990  RAC990 4321
***** Bottom of data *****
Scroll ==> PAGE

```

Figure 105. DFSMSrmm Product Details Panel

Data Set Details

DFSMSrmm displays a Data Set Details panel, such as the one shown in Figure 106.

```

Panel  Help
-----
EDGPD110                      DFSMSrmm Data Set Details
Command ==>

Data set name . . . : 'D023427.DSNAME'

Volume serial . . . : VOL190          Physical file sequence number . . . : 1
Owner . . . . . : MAZOWN22          Data set sequence number . . . : 0
Job name . . . . . : JOBNAME
Step name . . . . . : STEP01
Create date . . . . : 1996/010        DD name . . . . . :
Create time . . . . : 08:05:37        YYY/YYY/YYY Record format . . . . : FB
System id . . . . . : 9021           Block size . . . . . : 80
Date last read . . . : 1996/010       Logical record length : 80
Date last written . . :               Block count . . . . . : 5
Retention date . . . :               Unit address . . . . : 0BE0
VRS retained . . . . : NO            VRS management value. :
Matching VRS type . . :             Management class . . :
Matching VRS job name :             Data class . . . . . :
Matching VRS name . . :             Storage class . . . . :
                                   Storage group . . . . :

Security name . . . . :
Classification . . . . :

```

Figure 106. DFSMSrmm Data Set Details Panel

Vital Record Specification Details

If you specify a data set name, DFSMSrmm displays a Display Data Set vital record specification panel, such as the one shown in Figure 107.

Panel	Help

EDGPV110	DFSMSrmm Display Data Set VRS
Data set mask : '**'	GDG . . NO
Job name mask :	
Count . . . : 5	Retention type : CYCLES
	While cataloged : YES
Delay . . . : 0 Days	Until expired : NO
Location : OFFSITE	
Number in location : 5	
Priority : 0	
Next VRS in chain . : NXT1	Release options:
Chain using . . : AND	Expiry date ignore . . . : YES
	Scratch immediate : YES
Owner : WOODMW	
Description . . :	
Delete date . . : 1999/365	(YYYY/DDD)
Command ==>	

Figure 107. DFSMSrmm Display Data Set VRS Panel

If you specify a volume serial number, DFSMSrmm displays a Display Volume vital record specification panel, such as the one in Figure 108:

Panel	Help

EDGPV130	DFSMSrmm Display Volume VRS
Volume Serial : WOOD*	
Count . . . : 99999	
Delay . . . : 0 Days	
Location : OFFSITE	
Number in location : 99999	
Priority : 0	
Next VRS in chain . :	
Owner : WOODMW	
Description . . : Keep all master volumes off-site	
Delete date . . : 1999/365	(YYYY/DDD)
Command ==>	

Figure 108. DFSMSrmm Display Volume VRS Panel

If you specify a specific volume serial number, DFSMSrmm displays a simpler version of the Display Volume Vital Record Specification panel, because a specific volume cannot be kept in two stores simultaneously and can only have a count of one.

If you specify the name of a vital record specification, DFSMSrmm displays a Display Name vital record specification panel, such as the one in Figure 109.

Panel

Help

EDGPV140DFSMSrmm Display Name VRS

Name : NXT1

Count : 2

Retention type : CYCLES

While cataloged : NO

Until expired : NO

Location : HOME

Number in location : 2

Next VRS in chain . : NXT2

Chain using . . : NEXT

Owner : WOODMW

Description . . . :

Delete date . . : 1999/365 (YYYY/DDD)

Command ==>

Figure 109. DFSMSrmm Display Name VRS Panel

Displaying DFSMSrmm System Options

You can obtain information about your installation's parmlib options and DFSMSrmm control data set information using the DFSMSrmm ISPF dialog or RMM TSO LISTCONTROL subcommand.

To use the LISTCONTROL subcommand, access to the RACF profile STGADMIN.EDG.LISTCONTROL is needed. See the *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information.

Related TSO Subcommand

Use the LISTCONTROL subcommand with the OPTION operand to request information about system options defined by your installation. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information.

To display the panel shown in Figure 110 on page 172, you could enter a fast path command from the command or option line of any panel as shown in Figure 109.

```

Panel  Help  Scroll
-----
EDGPC200                      DFSMSrmm System Options Display

ParmLib Suffix . . : UK
Operating mode . . : PROTECT
More:

Data sets:
Control . . . : RMMTST.PERFORM.CDS
Journal . . . : JRNL1.CDS
CDS id . . . : DAYMVS1      Journal threshold . . . : 75 %
Catalog SYSID: *

Retention period:          SMF:
Default . . . : 8          System id . . : DAYMVS1
Maximum . . . : 9990       Audit . . . : 210
Catalog . . . : 12        hours Security . . . : 211

Report options:
Lines per page : 54
Date format . . : AMERICAN

Miscellaneous:              Auto-start procedures:
RACF support . . . . . : PREDEFINED  Scratch procedure : EDGXPROC
MAXHOLD limit . . . . . : 100        Backup procedure  : BACKPROC
IPL check . . . . . : NO
Uncatalog . . . . . : SCRATCH
System notify . . . . . : NO
TLCS V1 coexistence . . : NO
BLP . . . . . : RMM
Master overwrite . . . . : LAST
Message case . . . . . : MIXED
Accounting . . . . . : JOB
Disposition DD name . . : RMMDISP
Disposition message ID  : EDG4054I

VRS:
Job name . . . . . : 2
Minimum count . . . : 1
Minimum action . . : FAIL
Change . . . . . : INFO
VRSEL processing . . : NEW

Command ==>

```

Figure 110. DFSMSrmm System Options Display

Using Resource Lists

Any user can request a list of resources defined to DFSMSrmm. You use the DFSMSrmm ISPF panels or RMM TSO subcommands to define criteria for the list. You can change the sort order of some lists either before or after DFSMSrmm builds them. See “How to Sort a List” on page 176 for more information.

Depending on your level of authorization, you can enter line operators against items in some lists to request additional information or request certain functions. See “How to Use Line Operators” on page 182 for more information.

How to Request a List

Use the search panels or the RMM TSO SEARCH subcommands to request that DFSMSrmm display a list of resources. You can request lists of shelf locations, volumes, software products, data sets, and vital record specifications. See “Chapter 12. Using RMM TSO Subcommands” on page 221 for more information on the SEARCH subcommands.

Because DFSMSrmm allows you to specify generic names or numbers as search criteria, you can also use the search panels to get the specific name or number of a resource to request detailed information about it through the Display panels. For example, if all you know is a volume's owner you can use the search panels to request a list of all volumes owned by that owner. DFSMSrmm includes the volume serial number for each volume in the list, which you can then use to request detailed information about a specific volume on the Volume Display panel.

To request a list of resources defined to DFSMSrmm:

1. Select a Search panel. Table 15 describes how to select the various DFSMSrmm ISPF dialog search panels.

Table 15. How to Display Search Panels

To Display a Search Panel for:	Do One of the Following:
Shelf locations	<ul style="list-style-type: none"> • Select option 5 (RACKS) on the Librarian Menu, followed by option 4 (SEARCH) on the DFSMSrmm Rack and Bin Menu. • Select option 2 (RACKS) on the Command Menu, followed by option 4 (SEARCH) on the Rack and Bin Menu.
Volumes	<ul style="list-style-type: none"> • Select option 1 (VOLUME) on the Librarian Menu, followed by option 5 (SEARCH) on the Volume Menu. • Select option 1 (VOLUME) on the Administrator Menu, followed by option 5 (SEARCH) on the Volume Menu. • Select option 1 (VOLUME) on the User Menu.
Software products	<ul style="list-style-type: none"> • Select option 3 (PRODUCT) on the Librarian Menu. • Select option 5 (PRODUCT) on the Command Menu, followed by option 5 (SEARCH) on the Product Menu. • Select option 3 (PRODUCTS) on the User Menu.
Data sets	<ul style="list-style-type: none"> • Select option 3 (DATA SET) on the Command Menu, followed by option 5 (SEARCH) on the Data Set Search Menu. • Select option 2 (DATA SET) on the User Menu.
Vital record specifications	<ul style="list-style-type: none"> • Select option 3 (VRS) on the Administrator Menu, followed by option 5 (SEARCH) on the Vital Record Specification Menu. • Select option 6 (VRS) on the Command Menu, followed by option 5 (SEARCH) on the Vital Record Specification Menu.

2. Press ENTER.

DFSMSrmm displays a search panel, such as the Rack and Bin Search panel shown in Figure 111 on page 174:

Panel Help

EDGPR010 DFSMSrmm Rack and Bin Search

Command ==>

Specify one of:

Rack Number May be generic.

or

Bin Number May be generic.

Location SHELF SHELF, SMS library name, or storage location

Media name Limit to a single media name

Pool or to a particular pool

Status SCRATCH EMPTY, IN USE or SCRATCH

Limit Limit to first nnnn racks or bins. Default = 10.

Figure 111. DFSMSrmm Rack and Bin Search Panel

- Specify the search criteria you want DFSMSrmm to use to create the list. Press PF1 for field-specific help.

Table 16 lists the search criteria you can specify for each type of resource defined to DFSMSrmm:

Table 16. Search Criteria

For a List of:	You Can Search by:
Shelf Locations	Rack or bin number Location Status Media name or pool ID
Volumes	Volume serial number Volume owner Media name or pool ID Status Date assigned to a user or returned to scratch status
Software products	Software product number or name
Data sets	Data set name Volume owner or volume serial number Date data set was created Data set sequence number
Vital record specifications	Data set name mask or volume serial number

For example, to create a list of all empty shelf locations in the removable media library, use the Rack and Bin Search panel to specify a status of EMPTY, a location of LIBRARY, and leave the rack number field blank.

You can specify either specific or generic names or numbers on the search panels. This applies to rack and bin numbers, volume serial numbers, data set names, and software product number and names.

You can also specify how many entries you want DFSMSrmm to include in the list by entering a value in the Limit field.

To display pool IDs or media names for pools defined for your installation, type CONTROL VLPOOLS on the command or option line of any DFSMSrmm panel. You can also use the LISTCONTROL subcommand with the VLPOOL operand.

To display media names for storage locations defined for your installation, type CONTROL LOCDEF on the command or option line of any DFSMSrmm panel. You can also use the LISTCONTROL subcommand with the LOCDEF operand.

4. Press ENTER to build your list. You might have to scroll to the right on some lists to see all the information.

After DFSMSrmm builds your list, you can sort it using a different sort order. See “How to Sort a List” on page 176 for more information. You can also issue line operators against list items to access further information recorded by DFSMSrmm, or to request additional function. See “How to Use Line Operators” on page 182 for information on how to issue line operators against list entries.

Sometimes when you use generic names or numbers as search criteria, the resulting list of resources exceeds the storage space available to DFSMSrmm. DFSMSrmm returns as many list items as possible, and a message, such as the following example:

There is not enough storage available to list all the volumes.

You can do one of the following to obtain all available records from a search:

- Press PF3 to return to the panel where you specified the search criteria. Specify a generic volume serial number that limits the number of volumes to be searched.

For example, if you initially specified the following values on the Volume Search panel:

```
Volume      ==> 11*
Owner       ==> *
Limit       ==> *
```

you can divide the search by issuing several search requests using a more specific volume serial number mask each time.

For example, you could replace the generic volume serial number, 11* with three generic volume serial numbers VOLUME(111*), VOLUME(112*), and VOLUME(113*).

or

- Use one of the RMM TSO SEARCH subcommands instead of a panel to request the search and have the results come out in linemode messages. Use the SAVE ON command to process the subcommand in batch mode. See “Chapter 3. Getting Started with DFSMSrmm” on page 39 for information on the SAVE ON command.

How to Sort a List

You can change the sort order DFSMSrmm uses to build lists by:

- Using list sort options panels to change the default sort order
- Using the SORT command to change how a list is sorted as you view the list

The following sections describe how to use list sort options panels and the SORT command.

How to Change the Default Sort Order

In the dialog, you can display a sort options panel for each type of list. Use the Dialog Sort Options Menu to select the specific list sort options panel you need.

To display this menu, do one of the following:

- Select Option 2 (SORT) on the Dialog Options Menu and press ENTER.
- or
- Type **OPTIONS SORT** on the command line of any panel and press ENTER.

DFSMSrmm displays the Dialog Sort Options Menu as shown in Figure 112:

```
Panel  Help
-----
EDGP@OP2          DFSMSrmm Dialog Sort Options Menu
Option ==>

1 DATA SET - Specify Data Set list sort options
2 PRODUCT  - Specify Product list sort options
3 RACK     - Specify Rack list sort options
4 VOLUME   - Specify Volume list sort options
5 VRS      - Specify Vital Record Specifications list sort options

Enter selected option or END command.  For more info., enter HELP or PF1.

5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993
```

Figure 112. DFSMSrmm Dialog Sort Options Menu

1. Select the type of list that you want to sort.
2. Press ENTER. DFSMSrmm displays a list sort options panel corresponding to the type of list you specified.

For example, if you specify **1** and press ENTER, DFSMSrmm displays the Data Set List Sort Options panel as Figure 113 on page 177 shows:

Panel Help

EDGP@STD DFSMSrmm Data Set List Sort Options

Command ==>

Enter the relative sort priority (1 to 5) and direction (A or D):

Table field name	Priority	Direction
-----	-----	-----
Data set name		
Volume serial		
Owner		
File sequence		
Create date		

Enter END command to save changes, or CANCEL to end without saving.

Figure 113. DFSMSrmm Data Set List Sort Options Panel

On this panel, enter the sort priorities and sort directions of your choice, as follows:

1. Assign a priority level to any or all of the table field names available for data set lists. The highest priority is 1. If you do not specify a priority level, DFSMSrmm uses either the initial setting or the most recent value specified.
2. Assign a direction for any or all of the table field names to which you assigned a priority. Specify A to indicate ascending order, and D to indicate descending order. DFSMSrmm uses either the initial setting or the most recent value specified if you do not specify a sort direction.

Use all the list sort options panels in the same way. The only difference between these panels is the specific table field names used in the different lists.

Table field names correspond to *data columns* on lists. Each data column contains specific information recorded by DFSMSrmm for a particular resource. For example, DFSMSrmm records a rack number, media name, volume serial number, and rack status for each shelf location in the removable media library. Table field names differ between lists because DFSMSrmm records different information for different resources. Thus, each type of list is sorted differently. For example, only software product lists can be sorted by software product number.

Table 17 describes the different table field names you can sort on, and identifies the lists they are used in.

Table 17. Table Field Names on DFSMSrmm Lists

Table Field Name	Sort Command Parameter	Description	Type of List
Action	Action	Type of release action (E-erase, I-initialize, N-notify owner, O-return to owner, S-return to scratch, R-replace)	Volume action and movement
Assigned Date	Assigned	Date the volume was assigned to a user or returned to scratch status	Volume, volume action and movement
Attributes	Attributes	Special attributes associated with the volume (NONE, RDCOMPACT)	Volume
Cataloged	Cataloged	Whether or not the data set is cataloged (Y or N)	Vital record specification
Compaction	Compaction	Type of compaction technique (NONE, IDRC)	Volume
Create Date	Create	Date the data set was created	Data set
Data sets	Datasets	Number of data sets on the volume	Volume
Data Set Name	Dataset	Data set name	Data set
Delete Date	Delete	Date DFSMSrmm deletes the vital record specification	Vital record specification
Destination	Destination	Location where volume is moving to (HOME, LOCAL, DISTANT, REMOTE, <i>library_name</i>)	Volume, volume action and movement
Expiration Date	Expiration	Date the volume is to be considered for release	Volume, volume action and movement
Feature Code	Feature	Software product feature code	Software product
Home Location	Home	Home location name (SHELF, <i>library_name</i>)	Volume
Job	Job name mask	Vital record specification	Data set
Label	Label	Volume label type	Volume
Level	Level	Software product level (version and release)	Software product
Location	Location	Location where the volume resides	Volume, vital record specification, volume action and movement
Media Name	Medianame	Type of volume	Shelf location, volume
Media Type	Mediatype	Physical media type of volumes	Volume
Next VRS Name	Nextvrs	Name of next vital record specification to which this vital record specification is linked	Vital record specification

Table 17. Table Field Names on DFSMSrmm Lists (continued)

Table Field Name	Sort Command Parameter	Description	Type of List
Owner	Owner	Owner ID of the volume owner	Data set, volume, vital record specification, volume action and movement
Priority	Priority	Priority for a location	Vital record specification
Product Name	Name	Software product name	Software product
Product Number	Number	Software product number	Software product
Rack or Bin Number	Number	Shelf location in the removable media library or storage location	Shelf location
Rack Number	Number	Shelf location in the removable media library	Volume, volume action and movement
Recorded Format	Format	Type of recording format (18TRACK, 36TRACK)	Volume
Retention	Retention	Type of retention for data sets (DAYS, CYCLES, or LASTREFERENCEDAYS)	Vital record specification
Status	Status	Status of shelf location (EMPTY, INUSE, or SCRATCH) or volume (master, user, scratch, initialize, open, VRS, pending release, on loan)	Shelf location, volume
Sequence	Sequence	Data set sequence number	Data set
Transit	Transit	Status of volumes that have been ejected from a system-managed library but not yet confirmed as stored in their target destination	Volume, volume action and movement
Type	Type	Type of vital record specification (data set, volume, or name)	Vital record specification
Untilexpired	Untilexpired	Retention type	Vital record specification
Volume/Data Set/ Name Specification	Volume/Dataset/Name	Volume serial number, data set name, or vital record specification name	Vital record specification
Vols	Vols	Number of volumes associated with the software product	Software product
Volume Serial	Volume	Volume serial number	Data set, Software product, shelf location, volume, volume action and movement

How to Sort a List as You View It

Use the SORT command from the command line of any displayed list to sort the items in the list. The SORT command you use will vary depending on the type of list you are sorting: volume, action, data set, software product, rack or bin number, or vital record specification.

To specify the sort order for a list, type SORT followed by one or more parameter pairs, as follows:

```
SORT <direction field> . . .
```

where

direction is ASCENDING or DESCENDING, or an abbreviation

field is a data column name, or abbreviated name, for a specific type of list.

For example, if you type

```
SORT ASCENDING MEDIANAME
```

on the command line of a volume list, DFSMSrmm sorts your list of volumes by media name in ascending order.

You can use more than one pair of two-parameter sort specifications. For example, if you type

```
SORT ASCENDING MEDIANAME DESCENDING RACK
```

on the command line of a volume list, DFSMSrmm sorts your list of volumes first by media name type in ascending order, then by rack number in descending order.

The names you can use to specify data columns in lists depend on the type of list you are sorting. You can specify either a full column name or a unique abbreviation, as shown in the following syntax diagrams.

Rack and Bin Number Lists

Figure 114 describes the syntax to sort a list of rack or bin numbers:

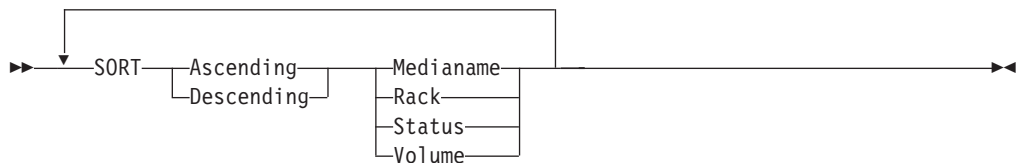


Figure 114. SORT Command Syntax Diagram for Rack or Bin Number Lists

Volume Lists

Figure 115 on page 181 describes the syntax to sort a list of volumes:

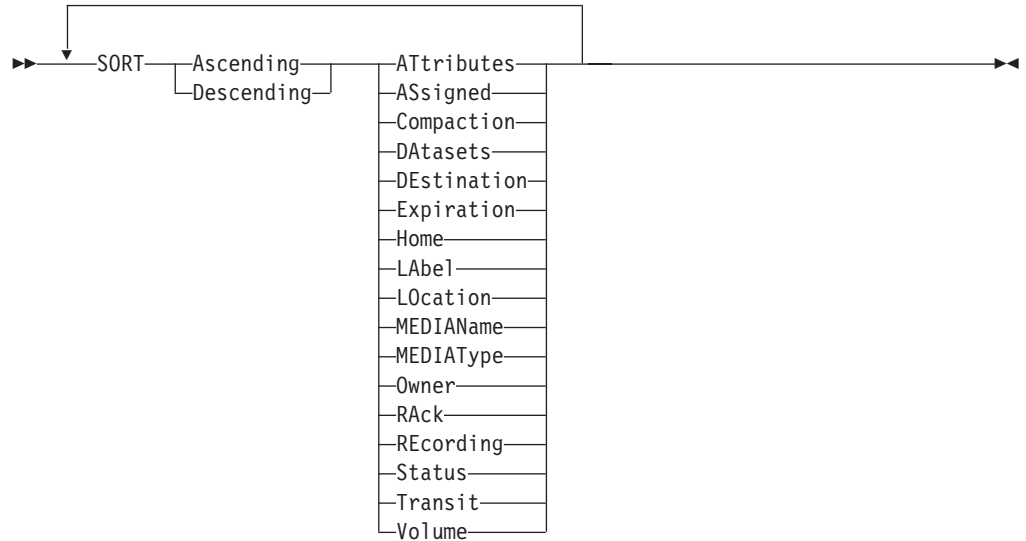


Figure 115. SORT Command Syntax Diagram for Volume Lists

Volume Movements and Actions Lists

Figure 116 describes the syntax to sort a list of volume movements and actions:

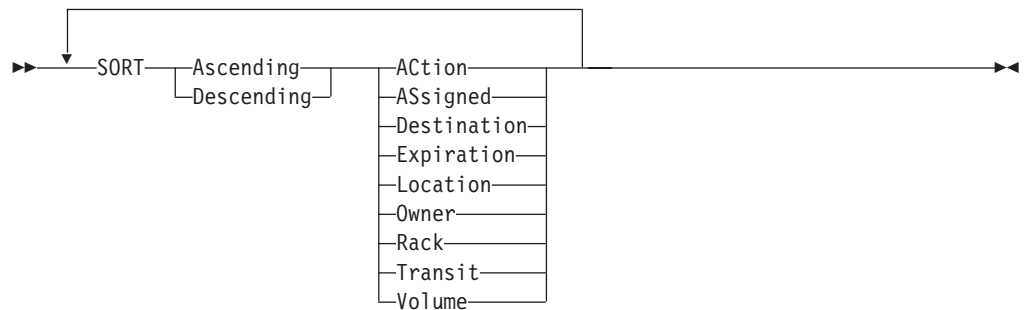


Figure 116. SORT Command Syntax Diagram for Lists of Volume Movements and Actions

Software Product Lists

Figure 117 describes the syntax to sort a list of software products:

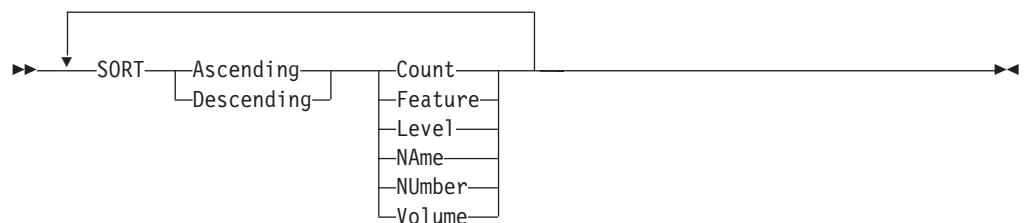


Figure 117. SORT Command Syntax Diagram for Software Product

Data Set Lists

Figure 118 describes the syntax to sort a list of data sets:

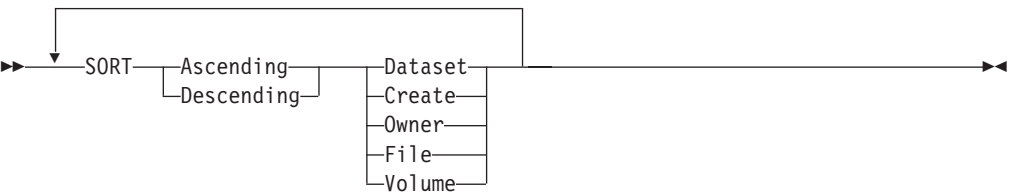


Figure 118. SORT Command Syntax Diagram for Data Set Lists

Vital Record Specification Lists

Figure 119 describes the syntax to sort a list of vital record specifications: Once you change the initial settings for sort values, DFSMSrmm stores the most

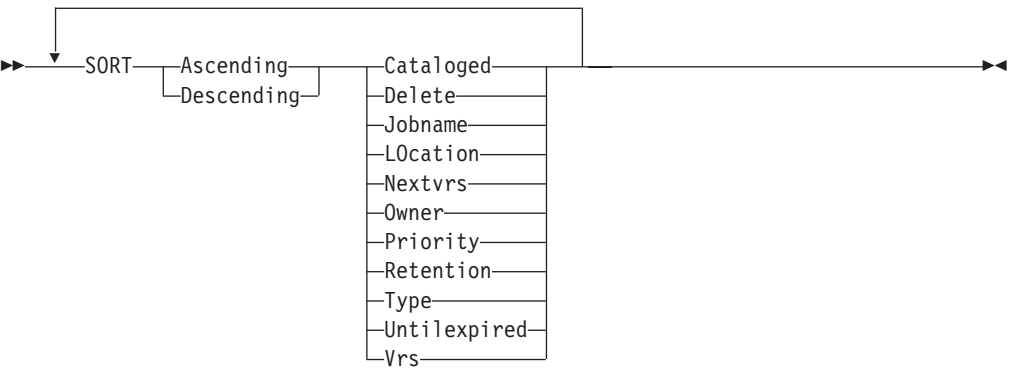


Figure 119. SORT Command Syntax Diagram for Rack or Bin Number Lists

recently specified values and remembers them across sessions.

DFSMSrmm uses the values you specify in the SORT command to prime the list sort options panels during your session.

How to Use Line Operators

You can issue line operators against entries in a list to display additional information, or to perform certain functions, such as changing or deleting existing information.

All DFSMSrmm users have read access to information stored by DFSMSrmm and can request lists of resources. However, to request any DFSMSrmm functions against one or more list entries, you must be an authorized user with read/write access to information in the list.

Shelf Location Lists

You can use the line operators shown in 182 to display information on shelf location lists.

Line Operator	Description
---------------	-------------

V	Display full volume information
VA	Display volume access information
VD	Display storage location information
VG	Display general volume information
VS	Display volume statistics

Volume Lists

You can use the line operators shown in 183 to perform functions on volume lists.

Line Operator

Description

C	Change volume information for an owned volume (See “Changing Volume Information” on page 60.)
I	Display data set name information
E	Eject volume from an IBM 3494 Tape Library Dataserver, IBM 3495 Tape Library Dataserver, or an IBM Model M10 3495 Tape Library Dataserver
F	Remove the volume from the removable media library, delete from DFSMSrmm, and uncatalog all data sets on the volume for which DFSMSrmm has recorded information in the control data set
L	List multi-volume chain
O	Display owner information
R	Release the owned volume. DFSMSrmm displays the DFSMSrmm Confirm Volume Release panel. DFSMSrmm processes any release actions specified for the volume, and marks as pending any release actions that must be done manually. Actions can be one or more of the following: <ul style="list-style-type: none"> • Return to scratch status • Replace the volume • Initialize the volume • Erase the volume • Notify the volume’s owner of the volume’s release • Return the volume to the owner

See “Releasing Volumes Manually” on page 158.

Software Product Lists

You can use the line operators shown in 183 to perform functions on software product lists.

Line Operator

Description

A	Add owned volume entries. DFSMSrmm displays the DFSMSrmm Change Product Details panel (See “Changing Software Product Information” on page 69.)
C	Change details for an owned software product (See “Changing Software Product Information” on page 69.)

D	Delete owned software product or associated volume (See “Deleting Software Product Information” on page 71.)
DN	Delete software product without releasing volume
P	Display software product details
R	Release software product volume
S	Complete details of selected entry for partial software product numbers.
V	Display volume details

Data Set Lists

You can use the line operators 184 to perform functions on data set lists.

Line Operator

Description

C	Change data set information (See “Changing Data Set Information” on page 73.) for an owned volume
D	Delete data set information for an owned volume. If you requested the confirm option for your session on the Dialog User Options panel, see “Setting Dialog Options” on page 41. DFSMSrmm displays the DFSMSrmm Confirm Delete Data Set. If you confirm deletion of the data set information, DFSMSrmm also deletes information for all subsequent data sets on the volume.
I	Display data set details
O	Display owner information
V	Display volume information

Vital Record Specification Lists

You can use the line operators shown in 184 to perform functions on vital record specification lists.

Line Operator

Description

C	Change a vital record specification
D	Delete a vital record specification
L	List the vital record specifications linked together in a chain
M	Find all matching resources
O	Display the owner of a vital record specification
S	Display a vital record specification

Requesting Specific Types of Lists

This section describes how to request specific types of lists using various panels and subcommands as well as the report extract data set.

Scratch Volume Pull Lists

Operators can use the RMM SEARCHRACK subcommand or the Rack and Bin Search panel to request a list of available scratch volumes to be pulled from their shelf locations and mounted for use.

Using the RMM SEARCHRACK Subcommand

To create a volume pull list, you can issue the RMM SEARCHRACK command as follows:

```
==> RMM SEARCHRACK RACK(*) SCRATCH LIMIT(nn)
```

where *nn* is the number of entries you want in the list. See “SEARCHRACK: Creating a List of Shelf Locations” on page 372 for more information.

You can produce a scratch pull list in batch, and send the output to a printer so that you have a hardcopy listing available to use when pulling tape volumes from the library.

The JCL shown in Figure 120 produces a list of up to 2000 scratch volumes from the pool C*. You can change the search parameters to produce a scratch pull list from the subset of all volumes in the library that you currently want to use for scratch processing.

```
// EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
RMM SEARCHRACK POOL(C*) SCRATCH LIMIT(2000)
/*
```

Figure 120. Producing a List of Scratch Volumes

Using the Rack and Bin Search panel

You can also use the Rack and Bin Search panel, shown in Figure 121 on page 186. Leave the RACK NUMBER field blank to list all scratch volumes. Specify SHELF to list all volumes residing in location SHELF. Specify the status as SCRATCH to limit the search to scratch volumes. Enter a limit to identify the number of entries you want in your list. The default is 10.

In addition, you can specify either a media name or a specific pool ID if you want to limit your list of scratch volumes to volumes of a certain type of media or from a certain pool.

Panel Help	

EDGPR010	DFSMSrmm Rack and Bin Search
Command ==>	
Specify one of:	
Rack Number	May be generic.
or	
Bin Number	May be generic.
Location SHELF	SHELF, SMS library name, or storage location
Media name	Limit to a single media name
Pool	or to a particular pool
Status SCRATCH	EMPTY, IN USE or SCRATCH
Limit	Limit to first nnnn racks or bins. Default = 10.

Figure 121. DFSMSrmm Rack and Bin Search Panel

DFSMSrmm displays a list of shelf locations containing scratch volumes. DFSMSrmm lists a specific rack number, media name, volume serial number, and status for each item in the list. Use this list to locate and pull the scratch volumes you need, then place them near the drives where you use them, or into cartridge loaders.

Drop/Ship Lists

When you move volumes between locations in the removable media library or between storage locations, use one of the following to request a drop/ship list:

- The SEARCHVOLUME subcommand
- The Volume Action Status panel

Using the RMM SEARCHVOLUME Subcommand

You can use the SEARCHVOLUME subcommand to create shorter lists than those built by EDGRPTD. For example, to request that DFSMSrmm build you a list of volumes to be moved to the LOCAL storage location, issue the SEARCHVOLUME subcommand, as follows:

```
==> RMM SEARCHVOLUME VOLUME(*) OWNER(*) LIMIT(*) DESTINATION(LOCAL)
```

DFSMSrmm displays the list you requested.

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information on the SEARCHVOLUME subcommand.

Using the Volume Action Status Panel

If you want to create a list of volumes to be moved and also confirm those moves to DFSMSrmm, use the Volume Action Status panel.

For example, to confirm that volumes have been moved from the removable media library to the LOCAL storage location, and to create a list of these volumes, use the Volume Action Status panel, as Figure 122 shows:

1. Type S against the type of volume move for which you want a list of volumes.

```

Panel  Help
-----
EDGPC700          DFSMSrmm Volume Action Status          Row 1 to 8 of 8
Command ==>          Scroll ==> PAGE

The following line commands are valid: C, U and S
Dest-
S Action  Location ination  Move Type Status
-----
ERASE                                PENDING
INIT                                PENDING
NOTIFY                                PENDING
REPLACE                                UNKNOWN
RETURN                                UNKNOWN
SCRATCH                                PENDING
S MOVE    LIBRARY  DISTANT  NOTRTS  PENDING
MOVE     REMOTE   LOCAL   NOTRTS  PENDING
***** Bottom of data *****

```

Figure 122. DFSMSrmm Volume Action Status Panel

2. Press ENTER.
DFSMSrmm returns a list such as the one in Figure 123:

```

Panel  Help
-----
EDGPT610          DFSMSrmm Volume Action Summary List      Row 1 to 4 of 4
Command ==>          Scroll ==> PAGE

The following line commands are valid: C, CE, CI, CM, CN, CO, CR, E, L, and V
Volume  Assigned  Expiration  Rack  Dest-  Tra-
S Serial Owner   Date       Date       Location Number Action ination nsit
-----
DMK000 ETZ001  1994/310   2019/092   LIB1    DMK000    LIB2    Y
DMK001 ETZ001  1994/310   1994/315   LIB1    DMK001    LIB2    Y
DMK002 ETZ001  1994/310   1994/360   LIB1    DMK002    LIB2    Y
DMK003 ETZ001  1994/310   1994/335   LIB1    DMK003    LIB2    Y
***** Bottom of data *****

```

Figure 123. DFSMSrmm Volume Action Summary List

3. Use the ISPF command, PRINT or PRINT-HI, to print the list.
You can use line operators to confirm moves. See “Confirming Volume Movements to DFSMSrmm” on page 124 for more information on confirming volume movements.

Lists of Pools Defined by Your Installation

Related TSO Subcommand

Use the LISTCONTROL subcommand with the VLPOOLS operand to request information about pools defined by your installation. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333 for more information.

Pools are groups of rack numbers with a common prefix. These rack numbers define shelf locations where you can store your volumes in logical groups. Pooling is a more effective way to manage your removable media library. You can define volume attributes at the pool level rather than just at the removable media library level.

To display information about the pools defined for your installation, do one of the following:

- Specify CONTROL VLPOOLS from the command or option line of any panel.
or
- Use the Control Information Menu. To view this menu, select one of the following:
Option 8 (CONTROL) on the Support Menu and press ENTER
or
Option 7 (CONTROL) on the Command Menu and press ENTER.

DFSMSrmm displays the Control Information Menu shown in Figure 124 on page 189:

```

Panel  Help
-----
EDGPC000          DFSMSrmm Control Information Menu
Option ==>

0 Options      - Specify dialog options and defaults
1 Control      - Display cds control record details
2 System       - Display system options and defaults
3 Security     - Display security classification rules
4 Vlpools      - Display volume pool definitions
5 Mntmsg       - Display mount message definitions
6 Reject       - Display volumes to be rejected
7 Actions      - Display volume moves and actions
8 Locdef       - Display location definitions

Enter selected option or END command.  For more info., enter HELP or PF1.
5695-DF1 (C) COPYRIGHT IBM CORPORATION 1993

```

Figure 124. DFSMSrmm Control Information Menu Panel

- Select Option 4 (VLPOOLS) on the Control Information Menu and press ENTER.

DFSMSrmm displays a Volume Pool Definitions panel, such as the one in Figure 125:

```

Panel  Help
-----
EDGPC400          DFSMSrmm Volume Pool Definitions          Row 1 to 4 of 4
Command ==>                                           Scroll ==> PAGE

Pool  Media   System  Pool  RA Ty Date  Description
      Name                Name   CF pe Check
-----
A20*  3480    D65MVS6          Y S NO   V
DEF30* 3480    D65MVS6          Y S NO   M
DEF31* 3480    D65MVS6          Y S NO   M
*      3480          N S OPER D
***** Bottom of data *****

```

Figure 125. Example of a List of Pools

Lists of Volume Movements and Actions

During vital record processing, or as part of releasing volumes, DFSMSrmm automatically performs many specific actions. Movements and some actions must be performed manually and then confirmed to DFSMSrmm before DFSMSrmm can

release volumes and continue with vital record management and release processing. You can request these moves and actions more easily by requesting that DFSMSrmm list them for you.

To create lists of volume movements and actions waiting to be performed use the Volume Action Status panel as shown in Figure 122 on page 187.

1. Type S against the type of volume move or release action for which you want a list of volumes.
2. Press ENTER.

DFSMSrmm returns a list such as the one in Figure 126:

Panel Help										
EDGPT610		DFSMSrmm Volume Action Summary List					Row 1 to 4 of 4			
Command ==>							Scroll ==> PAGE			
The following line commands are valid: C, CE, CI, CM, CN, CO, CR, E, L, and V										
Volume		Assigned		Expiration		Rack		Dest-		Tra-
S	Serial	Owner	Date	Date	Location	Number	Action	ination	nsit	

	DMK000	ETZ001	1994/310	2019/092	LIB1	DMK000		LIB2	Y	
	DMK001	ETZ001	1994/310	1994/315	LIB1	DMK001		LIB2	Y	
	DMK002	ETZ001	1994/310	1994/360	LIB1	DMK002		LIB2	Y	
	DMK003	ETZ001	1994/310	1994/335	LIB1	DMK003		LIB2	Y	
***** Bottom of data *****										

Figure 126. DFSMSrmm Volume Action Summary List

3. Use the ISPF command, PRINT or PRINT-HI, to print the list.
Table 18 describes the line operators you can use against entries in the list to confirm one or more of the outstanding moves or actions.

Table 18. Confirm List Line Operators

Line Operator	Description
C	Confirm any move or release action
CE	Confirm that the volume has been erased
CI	Confirm that the volume has been initialized
CM	Confirm that the volume has been moved.
CN	Confirm that the volume's owner has been notified
CO	Confirm that the volume has been returned to its owner
CR	Confirm that the volume has been replaced
L	List multi-volume chain

See “Confirming Volume Movements to DFSMSrmm” on page 124 for more information.

CLISTs of Executable Subcommands

You can use any of the RMM TSO SEARCH subcommands with the CLIST operand to create a data set of executable subcommands for resources that you can modify and issue at a later time. For example, you can create a list of subcommands to confirm volume moves or release actions that have been completed, or a list of volumes moving out of a specific automated tape library dataserwer that must be ejected.

The search values you specify on the SEARCH subcommands create the records in the file. The CLIST values you specify add subcommands and operands to those records. You specify CLIST values in prefix and suffix text strings. If the total length of each text string exceeds 255 characters, the strings are split into multiple records and DFSMSrmm adds a continuation character, +, to all but the last record. Enclose your text string in quotes if the string contains any separator characters, such as blanks and commas. Use a blank or a comma between the text strings when you specify both a prefix and a suffix value. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the *prefix_string* with the data DFSMSrmm returns and the *suffix_string*.

For example, if you specify:

```
RMM SEARCHVOLUME VOLUME(*) LOCATION(LIB1) CLIST('RMM LV ',' STORE') -  
OWNER(*) LIMIT(*)
```

DFSMSrmm creates a data set of LISTVOLUME subcommands that request storage location information for each volume residing in the library, LIB1.

See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more examples of using the SEARCHVOLUME subcommand with the CLIST operand.

Printing Resource Information

To print a displayed list or a details panel, use the ISPF command PRINT and PRINT-HI.

Use the PRINT command to print panels a screen at a time and save the output in your ISPF list file. If the displayed information exceeds one screen, you must scroll to the next screen of entries and issue the PRINT command for each screen of entries.

Use the PRINT-HI command to print panels so that any highlighted characters on a panel appear in bold print.

You can assign either PRINT or PRINT-HI to a PF key. See *OS/390 ISPF User's Guide, SC28-1239* for more information.

Chapter 11. DFSMSrmm Operator Procedures

This chapter contains information for:

- Operators who might need to stop and restart the DFSMSrmm subsystem, respond to operator messages, and manually erase and initialize tape volumes
- Storage administrators or tape librarians who develop your installation's operations procedures, including responses to operator messages

Restarting the DFSMSrmm Subsystem

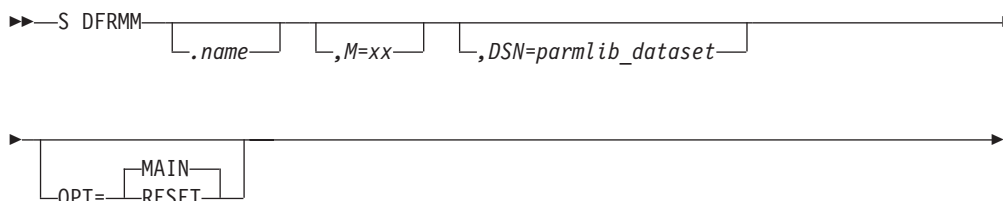
Under normal circumstances, the DFSMSrmm subsystem starts automatically through IPL procedures, either standard or as modified by your installation. In exceptional cases, such as after recovery of the DFSMSrmm control data set, you might need to restart the subsystem.

The following sections describe how to use the START command to restart the DFSMSrmm subsystem, and how to restart the subsystem after subsystem parameters have been modified.

Using the START Command

Use the START command to restart the DFSMSrmm subsystem as shown in Figure 127.

Figure 127. START Command Syntax Diagram



where:

DFRMM

DFRMM is the default procedure name. Your installation might have defined a name other than DFRMM. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on defining the started procedure name.

.name

Specifies a name other than the defined procedure name (DFRMM in the syntax diagram above) by which you can call the started procedure. You can then use this name on STOP and MODIFY commands. See *OS/390 MVS Planning: Operations* for information.

,M=xx

xx specifies a parmlib member name suffix with which DFSMSrmm must be started instead of the default parmlib member. Contact your system programmer to find out what values to use.

,DSN=parmlib_dataset

Specifies an alternative data set name to be used to find the parmlib member for this restart of the DFSMSrmm subsystem.

OPT=

Use OPT= to start or reset the subsystem interface.

MAIN

Use OPT=MAIN to start and activate the DFSMSrmm subsystem address space. OPT=MAIN is the default.

RESET

Use OPT=RESET to reset the DFSMSrmm subsystem interface to remove DFSMSrmm from the system.

At the console, enter S DFRMM to start DFSMSrmm, using the default procedure name and parmlib member.

When the DFSMSrmm subsystem interface is enabled, tape volume usage is rejected until the DFSMSrmm subsystem is active.

Implementing Subsystem Parameter Changes

The DFSMSrmm subsystem parameters can be changed anytime so that DFSMSrmm can use any of the installation options specified by the member name EDGRMMxx. For example, new pools or new classification definitions can be set through EDGRMMxx options. After changes have been made to the parmlib, you must restart the DFSMSrmm subsystem to implement the changes.

To restart DFSMSrmm and implement new parmlib options, use the command shown in Figure 128:

```
F DFRMM,M=xx
```

Figure 128. Command Syntax for Restarting DFSMSrmm and Implementing New Parmlib Options

In this example, DFRMM is the DFSMSrmm default procedure name. Your installation might have defined a different name. You could also use the name you used on the START command with the *name* parameter. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information on defining the procedure name and on setting EDGRMMxx parmlib options.

The DFSMSrmm subsystem temporarily stops and reinitializes itself with the new options. Before stopping, DFSMSrmm completes any requests that it is processing. New and queued requests are not processed until reinitialization is completed

To refresh the DFSMSrmm installation exits and enable any exits which are currently disabled, issue the command shown in Figure 129.

```
F DFRMM,REFRESH EXITS
```

Figure 129. Command Syntax for Refreshing DFSMSrmm Installation Exits

You will probably use this command at the request of the systems programmer to overcome a problem with the current installation exits.

Stopping the DFSMSrmm Subsystem

Before you can shut down the DFSMSrmm subsystem, you must wait until all current requests are completed and any outstanding requests are flushed from the request queues. All jobs that are processing and using tapes should be completed before you stop the subsystem to allow DFSMSrmm to record details of tape usage when a tape data set is closed. Before stopping the DFSMSrmm subsystem, stop the batch initiators to avoid failing a job that opens a tape data set. When you stop DFSMSrmm, it cannot successfully shutdown if another address space is using DFSMSrmm resources. For example, there might be an DFSMSrmm WTOR outstanding for a batch job. If DFSMSrmm shutdown is delayed, DFSMSrmm issues a message to inform you of the delay.

Except when DFSMSrmm is being shut down for recovery or formal changes to the MVS system, do not shut DFSMSrmm down unless you are quiescing the entire operating system. If DFSMSrmm is not present or active to validate tape requests, all tape usage is prevented to maintain volume integrity.

In some recovery situations, you might need to remove DFSMSrmm from the system. Use the DFSMSrmm EDGRESET utility to remove DFSMSrmm from the system and allow tape mounts to be processed without a media management system. If you do this, you must keep a record of all tape activity that occurs during this period so you can update the DFSMSrmm control data set at a later time. If you are collecting SMF records for tape activity, you could use these as a source of information for updating the control data set.

The following command examples use the default procedure name DFRMM. Your installation might have defined a different name. You could also use a name you specified on the START command with the *name* parameter.

To shut down the DFSMSrmm subsystem, use the MVS stop command, as shown in Figure 130:

```
P DFRMM
```

Figure 130. Command Syntax for Shutting Down DFSMSrmm

Before stopping, DFSMSrmm processes all existing requests to completion. New requests are rejected. If an existing, queued request is for inventory management, it fails.

To manually quiesce the DFSMSrmm subsystem, use the MVS MODIFY command, as shown in Figure 131:

```
F DFRMM,QUIESCE
```

Figure 131. Command Syntax for Quiescing DFSMSrmm

When you issue the quiesce command, DFSMSrmm completes any requests being processed. Queued request are not processed until you issue a command to take DFSMSrmm from the quiesced state and reinitialization is completed. If you stop DFSMSrmm from the quiesced state and any requests are outstanding, message EDG1107D prompts you with your choices of action.

To remove DFSMSrmm, issue the command shown in Figure 132:

```
S DFRMM,OPT=RESET
```

Figure 132. Command Syntax for Removing DFSMSrmm Using the EDGRESET Utility

You can use the RESET option even if your installation does not have RACF or an equivalent security product installed,

You can restart the DFSMSrmm subsystem using appropriate options for running either without tape recording or with tape recording and validation. You do not need to IPL to revert back to full DFSMSrmm subsystem function.

Controlling the Problem Determination Aid (PDA) Facility

The PDA facility default operating mode is trace enabled at DFSMSrmm start-up.

PDA tracing should be continuously enabled when DFSMSrmm is active since the processing overhead is minimal. If the EDGPDOX and EDGPDOY DD statements are defined in the DFSMSrmm start-up procedure, the EDGPDOX and EDGPDOY data sets are swapped and trace output is logged in the data sets.

PDA tracing can be controlled by using the following MODIFY command keywords. PDA processing can also be enabled or disabled by using the parmlib OPTION command PDA operands described in *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

PDA=ON|OFF

Specify to turn PDA tracing on or off.

PDALOG=ON|OFF|SWAP

Specify to control the LOGGING function during PDA tracing.

This example shows how to use the MODIFY command to enable PDA tracing.

```
F DFRMM,PDA=ON
```

The PDA log data sets are automatically swapped at DFSMSrmm start-up. Although there is no way to control swapping at start-up, you can use the MODIFY command PDALOG=SWAP to manually swap the data sets as required. For information on how to manually swap the PDA log data sets, see the *DFSMS/MVS DFSMSrmm Diagnosis Guide* manual for details.

Responding to DFSMSrmm Operator Messages

DFSMSrmm issues several messages requiring operator intervention. Your installation might have developed procedures you should follow in responding to these messages.

You might want to pay special attention to the messages listed in Table 19 on page 197, identified by type and number:

Table 19. Operator Messages

Type of Message	Message Numbers
Initialization messages	EDG0001I, EDG0204I, EDG0110D, EDG0114I, EDG0105I, EDG0101I, EDG0104E, EDG1001A, EDG0103D, EDG0123D, EDG0215D, EDG1101I, EDG0115D, EDG1105I, EDG1106I, EDG1107D, EDG9115I
Tape mount messages	EDG6627A, EDG6628A, EDG6663D, IEF233A, EDGV01D
Tape validation messages	EDG4021I, EDG4023I, EDG4024I, EDG4025I, EDG4026I, EDG4027I, EDG4028I, EDG4032I, EDG4033I, EDG4035I, EDG4036I, EDG4041I
Tape processing messages	EDG4008A, EDG4001D, EDG4010D, EDG4007E, EDG4005E, EDG4006E, EDG4012D
System error messages	EDG0154I, EDG2103D, EDG2104E, EDG2107E, EDG2108E, EDG2110I, EDG2111I, EDG2112I, EDG2113I, EDG2114I, EDG2115I, EDG2116A, EDG9116I
Installation exit messages	EDG0300I, EDG0301I, EDG0302I, EDG0303D, EDG0304I, EDG0305I, EDG8121D
Tape Labeling Messages	EDG6619I, EDG6620I, EDG6621E, EDG6622I, EDG6623I, EDG6625I, EDG6626A, EDG6627A, EDG6628A, EDG6631I, EDG6642I, EDG6658I, EDG6661E, EDG6662E, EDG6663E, EDG6672E

See *OS/390 MVS System Messages, Vol 2 (ASB-EWX)* for a complete description of the DFSMSrmm messages.

Initialization Messages

DFSMSrmm issues several types of initialization messages requiring your intervention. Your installation should have set procedures, many of them automated, for handling many of the error situations described in these messages.

Critical initialization errors usually cause DFSMSrmm to shut down. Report any diagnostic error messages to your system programmer as soon as possible.

If a working version of the old initialization parameters is still available, you can restart DFSMSrmm using the old parameters. If tapes must be used and the DFSMSrmm subsystem cannot be started, you can remove DFSMSrmm from the system and, if your installation permits, allow tape mounts to be processed without a media management system. See “Stopping the DFSMSrmm Subsystem” on page 195 for more information.

The following lists initialization messages, many that require your response:

**EDG0001I DFSMSrmm SUBSYSTEM INTERFACE INITIALIZATION
COMPLETE FOR ENTRY *ssname***

- DFSMSrmm is beginning to come up. It is still down. Wait for message EDG0105I. You do not need to reply to this message.

EDG0204I DFSMSrmm BEING INITIALIZED FROM MEMBER *member_name* IN *parmlib_dataset*

- DFSMSrmm is getting setup information from its parmlib member. Wait for message EDG0105I. You do not need to reply to this message.

EDG0110D ENTER TODAY'S DATE WITH FORMAT *day date_option* OR "CANCEL"

- Reply with today's date. Use the date format set by your installation. where:

- *day* is MON, TUE, WED, THU, FRI, SAT, SUN
- *date_option* is the date format set by your installation

For example, reply with: THU,02/20/1993.

- This reply is checked against the system date, and the message reoccurs if the system date is different. In this case, either reset the system date or reply with the correct current date.

Do not automate your response to this message. Until you respond to this message, DFSMSrmm initialization does not complete and no tape mounts are allowed.

EDG0114I SYSTEM DATE *system_date* VERIFIED

- The date entered in message EDG0110D is the same as the system date. You do not need to reply to this message.

EDG0105I DFSMSrmm SUBSYSTEM INITIALIZATION COMPLETE

- DFSMSrmm is now up and active. You do not need to reply to this message.

EDG0101I STARTED TASK ENDED BECAUSE THE DFSMSrmm SUBSYSTEM IS ALREADY ACTIVE

- DFSMSrmm is already up and active, but a START command was issued again and the second START command failed.

You do not need to reply to this message since DFSMSrmm is already running. Do not start DFSMSrmm without first stopping DFSMSrmm.

EDG0104E DFSMSrmm SUBSYSTEM INITIALIZATION FAILED

- DFSMSrmm is down and could not be brought up. Notify your system programmer immediately. Do not mount any tapes.

You do not need to reply to this message.

EDG1001A DFSMSrmm ADDRESS SPACE ENDING - RESTART IT

- DFSMSrmm is down.
- Start DFSMSrmm. If it does not come up, notify your system programmer immediately. Do not mount any tapes.

You do not need to reply to this message.

EDG0103D DFSMSrmm SUBSYSTEM INTERFACE IS INACTIVE - ENTER "IGNORE", "CANCEL", OR "RETRY"

- DFSMSrmm issues this message at DFSMSrmm procedure start-up time or at refresh time when DFSMSrmm is modified to use a new parmlib member.
- Reply RETRY.

EDG0123D INVENTORY MANAGEMENT FOUND TO BE ACTIVE ON SYSTEM *system_name* - REPLY "Y" TO RESET STATUS OR "N"

- DFSMSrmm issues this message if inventory management is trying to run on two shared systems, but only one system should run it. DFSMSrmm also

issues this message if the subsystem has determined that inventory management is in process on another system but has not completed successfully. This could occur if an MVS system has failed and recovery is in process on another system.

- Notify your system programmer and reply as directed.

EDG0215D ERRORS DETECTED IN INITIALIZATION PARAMETERS - ENTER "Y" TO CONTINUE OR "N" TO CANCEL.

- Errors occurred while bringing up DFSMSrmm, which generate other error messages.
- Check for other error messages. Based on those messages, either reply Y to allow DFSMSrmm to run or reply N to bring down DFSMSrmm.
- Notify your system programmer immediately if you bring down DFSMSrmm, as this causes all tape mounts to fail.

EDG1101I DFSMSrmm *command* COMMAND ACCEPTED

- A stop or modify command was issued and completed. You do not need to reply to this message.

EDG0115D THE DFSMSrmm SUBSYSTEM IS NOT RUNNING UNDER A JOB ENTRY SYSTEM - SOME DFSMSrmm FUNCTIONS ARE NOT AVAILABLE. REPLY "IGNORE" OR "CANCEL"

- DFSMSrmm issues this message at DFSMSrmm procedure start-up time or at refresh time when DFSMSrmm is modified to use a new parmlib member. The DFSMSrmm NOTIFY function is one of the functions that is not available when DFSMSrmm is not running under a JES2 or JES3 subsystem.
- Notify your system programmer and reply as directed.

EDG1105I STOP COMMAND ENTERED WHILE DFSMSrmm IS QUIESCED AND REQUESTS ARE WAITING TO BE PROCESSED

- DFSMSrmm issues this message if the STOP command is entered and DFSMSrmm is already quiesced.
- Reply to message EDG1107D as directed.

EDG1106I STOP COMMAND ENTERED WHILE DFSMSrmm IS QUIESCED AND REQUESTS ARE WAITING TO BE PROCESSED - INCLUDING CATALOG STATUS TRACKING

- DFSMSrmm issues this message if the STOP command is entered and DFSMSrmm is already quiesced.
- Reply to message EDG1107D as directed.

EDG1107D REQUESTS WAIT TO BE PROCESSED – REPLY "STOP", "QUIESCE", "RESTART", OR "M=xx"

- DFSMSrmm issues this message if the STOP command is entered and DFSMSrmm is already quiesced.
- Reply to message EDG1107D as described in Table 20 on page 200.

Table 20. Operator Responses to Message EDG1107D

To	Reply	And the result is
Stop DFSMSrmm	STOP	DFSMSrmm fails the waiting requests and stops.
Return to the quiesce state	QUIESCE	DFSMSrmm returns to the quiesce state.
Restart DFSMSrmm and process waiting requests	RESTART	DFSMSrmm is restarted using the current DFSMSrmm EDGRMMxx parmlib member.
Restart DFSMSrmm using a specific parmlib member and process waiting requests	M=xx	DFSMSrmm is restarted using the DFSMSrmm EDGRMMxx parmlib member you specified.

EDG9115I I/O INHIBITED FOR DFSMSrmm PROBLEM DETERMINATION OUTPUT DATA SET, REASON CODE *reason_code*

- DFSMSrmm issues this message at start-up time if the PDA trace EDGPDOX and EDGPDOY data sets are not defined to DFSMSrmm. DFSMSrmm issues this message once when EDGPDOX is not defined and then again if EDGPDOY is not defined.
- You need not respond to this message.

Tape Mount Messages

DFSMSrmm displays tape fetch and mount messages on both the operator console and the drive display panel. DFSMSrmm modifies fetch and mount messages to provide information about shelf location and scratch pools, when appropriate.

To list mount messages at the console, you can enter

```
"D R,KEY=MOUNT"
```

at the console.

Use MNTMSG commands in parmlib to identify the messages to be updated. DFSMSrmm adds a pool prefix to the message for a non-specific mount request and to the tape drive display unless you specify a pool name using the VLPOOL commands with the NAME operand. In that case, DFSMSrmm adds the pool name to the message. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information on the DFSMSrmm parmlib options.

For a specific mount request, DFSMSrmm adds the rack number to the message text. The rack number identifies the specific shelf location where the volume resides to help you locate the volume. This is particularly useful in those cases where a volume's rack number differs from its volume serial number.

When the rack number or pool prefix or pool name is added to the end of the mount message, the rack number or pool identifier is preceded by either RACK= or POOL=. When the rack number or pool identifier is added anywhere else in the message, DFSMSrmm does not add the descriptive text.

Following are some of the mount messages DFSMSrmm issues:

EDG6627A M *drive_number* **VOLUME(volser)** **RACK(rack_number)** **TO BE**
erased_or_labelled_or_verified, label_type

- Mount the volume, write enabled. You do not need to reply to this message. DFSMSRmm returns a null *rack_number* when the volume is not defined to DFSMSRmm.
- If the volume cannot be mounted, reply 'S' to skip the volume.

EDG6628A *drive_number* REPLY WITH RACK NUMBER OR VOLUME SERIAL NUMBER FOR NL VOLUME

- The volume does not have a volume label, so DFSMSRmm cannot verify that the correct volume is mounted. Check that the correct volume is mounted. This message is issued during EDGINERS processing.
- Reply with the volume serial number or rack number.

EDG6663D REPLY "R" TO RETRY OR "F" TO FAIL THE REQUEST, OR "A" TO ACCEPT THE MOUNTED VOLUME

- During EDGINERS processing, DFSMSRmm detected an incorrect volume serial number on a mounted volume. The EXEC statement PARM WRONGLABEL parameter has been specified and DFSMSRmm prompts the operator to reply. DFSMSRmm issues this message after issuing either message EDG6661E or EDG6662E.
- Reply as described in Table 21.

Table 21. Operator Responses to Message EDG6663D

To	Reply	And the result is
Accept the mounted volume	A	DFSMSRmm relabels the volume to the new volume serial number.
Fail the request	F	DFSMSRmm unloads the mounted volume and the request fails.
Retry to request	R	DFSMSRmm unloads the mounted volume and reissues the mount request message EDG6627A.

**IEF233A M *ddd*,*volser*|*pool*{,SL|NL}, *jobname*,|*jobname*,*stepname*|
RACK=*rackno*|POOL=*poolid***

- This message is the normal mount message with addition of a rack number or pool ID. Use the rack number or the pool ID to locate the volume.
- The variables in the message text are:
 - jobname* is the job.
 - stepname* is the step.
 - poolid* can be a pool prefix which is the starting characters of a rack number followed by asterisks (*) or a pool name which can be any one to eight character name your installation defines.
 - rackno* is the external tape label.
 - volser* is the internal tape label for SL tapes.

For example:

```
IEF233A M BE1,PRIVAT,SL,PRITCDX,WRIT1,PRITCDS.POOL2.TEST - POOL=SCRATCH
```

shows the pool name added to the end of a message.

The following message shows the pool prefix updated in the existing message text:

```
IEF233A M BE1,A01***,SL,PRITCDX,WRIT1,PRITCDS.POOL2.TEST
```

When DFSMSrmm uses the pool name rather than the pool prefix to update the tape drive display for a mount request, if the pool name is six characters or less, DFSMSrmm only updates the middle six characters of the display.

The drive display of eight characters normally looks like this for a mount of a standard label non-specific volume:

MPRIVATS

When DFSMSrmm updates the display with a six character pool prefix, the display might look like this:

MA01***S

When DFSMSrmm updates the display with an eight character pool name, the display might look like this:

BLUE-TAG

When DFSMSrmm updates the display with a pool name that is six characters or less, the display might look like this:

MRED S

DFSMSrmm provides EDG019VM as a replacement for IFG019VM. This installation exit is used to obtain the tape label of an NL tape mounted as a scratch tape on a non-specific request from the operator.

**EDGV01D dev REPLY WITH RACK NUMBER FOR NL REQUEST — OR
REPLY "REJECT"**

- DFSMSrmm issues this message to request the rack number for the NL scratch tape to be mounted.
- Reply with the volume serial number for the scratch tape.

Tape Validation Messages

At tape mount time, DFSMSrmm performs additional checking, which might result in a tape being rejected. See "How Does DFSMSrmm Validate Tape Mounts?" on page 17 for more information.

A tape rejection message is issued to explain why the tape was rejected and to help you decide what to do next. If a tape is rejected for a specific volume request, the user's job ends abnormally with F13, F14, or F37. The user receives a message indicating the reason for the rejection, and should refer to the message explanation for help in solving the problem.

DFSMSrmm can reject specific mount requests depending on the OPTION BLP(NORMM/RMM) that is specified in the EDGRMMxx parmlib member.

For a specific mount request of a volume not defined to DFSMSrmm, the volume is rejected only if the REJECT option in effect is either REJECT ANYUSE(*) or REJECT OUTPUT(*). The job then ends abnormally.

If a tape is rejected for a non-specific volume request, the mounted tape is unloaded and the system mount message is reissued. File the rejected tape on a separate shelf until the reason for its rejection can be determined. Usually a tape is rejected because of a mismatch of user JCL specifications, tape header labels and DFSMSrmm control data set data. In most cases, you must obtain a listing of the

volume's header labels to provide the tape librarian with sufficient diagnostic information. Use JES3 Tape Display DSP, MVS/DITTO or an equivalent to get this information.

See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on the type of checking DFSMSrmm performs when mounting tape volumes.

Following are some of the tape validation messages DFSMSrmm issues:

EDG4021I VOLUME *volser* REJECTED. IT IS NOT IN A SCRATCH POOL

- The volume is not assigned to a scratch pool and only volumes from scratch pools can be used for scratch tape mounts. This message is followed by messages EDG4005E or EDG4006E.

EDG4023I VOLUME *volser* REJECTED. IT MAY NOT BE USED ON MVS SYSTEMS

- The volume cannot be used on an MVS system. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4024I VOLUME *volser* REJECTED. BLP OUTPUT IS NOT PERMITTED TO SCRATCH OR MASTER VOLUMES

- Bypass label processing (BLP) is being used to write data to a master or scratch tape volume. This is only allowed on volumes which are in USER status if OPTION BLP(RMM) is set in parmlib member EDGRMMxx. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4025I VOLUME *volser* REJECTED. READING OF SCRATCH VOLUMES IS NOT PERMITTED

- The volume has been returned to scratch status, so it can only be written to as a new tape.
- The volume is demounted. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4026I VOLUME *volser* REJECTED. DATA SET NAME ON FIRST FILE DOES NOT MATCH RECORDED NAME

- The DFSMSrmm control data set has a different data set name for the first file on the volume. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4027I VOLUME *volser* REJECTED. IT IS NOT A SCRATCH VOLUME AND MOUNT REQUEST WAS NON-SPECIFIC

- A non-scratch volume was mounted but a scratch tape was requested.
- The volume is demounted. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4028I VOLUME *volser* REJECTED. VOLUMES WITH NON STANDARD LABELS ARE NOT SUPPORTED

- DFSMSrmm has detected a volume with a label type request of NSL, which is not supported. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4032I VOLUME *volser* REJECTED. IT HAS EXPIRED AND IS PENDING RELEASE

- The volume has expired but has not been released.
- The volume is demounted. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4033I VOLUME *volser* REJECTED. THE VOLUME IS WAITING TO BE REINITIALIZED

- The volume must be initialized before it can be used.
- The volume is demounted. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4035I VOLUME *volser* REJECTED. VOLUME IS SCRATCH AND OUTPUT NOT TO FIRST FILE

- A file other than the first file was written to, but the volume is a scratch volume. For scratch volumes, the first file must be written to first.
- If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4036I VOLUME *volser* REJECTED. REQUEST WAS FOR A SPECIFIC SCRATCH VOLUME

- The volume is a scratch volume and cannot be specifically requested by name.
- The volume is demounted. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4041I VOLUME *volser* REJECTED. DATA SET NAME DOES NOT MATCH FOR A MASTER VOLUME

- A data set name was specified that does not match information about the volume in the DFSMSrmm control data set.
- The data set is not overwritten. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4048I VOLUME *volser* REJECTED. MOUNTED VOLUME DOES NOT EQUAL REQUESTED VOLUME

- DFSMSrmm only allows the mounted and requested volumes to be different if neither volume serial number is defined to DFSMSrmm. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

EDG4049I VOLUME *volser* REJECTED. OPERATOR REQUESTED "CANCEL" FOLLOWING FAILURE OF INSTALLATION EXIT

- The processing of an OPEN request for a tape volume failed because of a failure in an DFSMSrmm installation exit. If DFSMSrmm is operating in warning mode, the volume indicated in this message can be used and DFSMSrmm issues message EDG4004I. If DFSMSrmm is not operating in warning mode, DFSMSrmm issues message EDG4005E or EDG4006E.

Tape Processing Messages

Following are some of the messages DFSMSrmm issues during tape processing. These messages describe the operator action that is required. You might want to set up handling procedures for some of them.

The first messages in the list, EDG4001D and EDG4010D, describe errors that can occur before tape validation occurs.

EDG4001D DFSMSrmm I/O ERROR DURING *open_or_close_or_eov* FOR *volser* BY *jobname, stepname, ddname*; ENTER "RETRY" OR "CANCEL"

- Notify your system programmer immediately that the DFSMSrmm control data set must be recovered.

EDG4010D BACKUP IN PROGRESS DURING *open_or_close* FOR *volser* BY *jobname, stepname, ddname* ; ENTER "RETRY" or "CANCEL"

- Wait until backup is completed before continuing processing. If you get the same message, wait again before replying.
- Contact your system programmer if you cancelled tape processing so that corrective action can be taken.
- If you don't know if the backup is completed, enter retry again. If backup is still in progress, try again later.

EDG4008A SECURE *security_number* VOLUME *volser* IN USE BY *jobname, stepname, ddname* REPLY WHEN READY

- This message informs you that the requested volume is a secure volume, requiring specific handling procedures.
- Follow your installation's procedures for such cases before replying to this message.

EDG4007E THE DFSMSrmm SUBSYSTEM IS NOT ACTIVE, USE OF *volser* BY *jobname, procname, stepname, ddname* REJECTED

- Check outstanding replies to see why DFSMSrmm is not active. If necessary, start DFSMSrmm. See "Restarting the DFSMSrmm Subsystem" on page 193 for information on restarting DFSMSrmm.
- IEC502E and IEC501A messages follow this message as DFSMSrmm cancels the mount request.

EDG4005E VOLUME *volser* ON *unit_address* REJECTED FOR USE BY *jobname, stepname, ddname*

- The volume cannot be used to satisfy this mount request. Look for another message with further information.
- If this is a scratch volume, mount another scratch volume; otherwise cancel the job using information from the previously issued EDG message.

EDG4006E VOLUME *volser* ON *rack_number* REJECTED FOR USE BY *jobname, stepname, ddname*; OPEN REQUEST FAILED BY DFSMSrmm

- Neither this nor any other volume can be used for this mount request
- The tape is rejected and the job abnormally ends.

EDG4012D THE DFSMSrmm SUBSYSTEM IS NOT ACTIVE DURING
open_or_close **FOR** *volser* **BY** *jobname, procname, stepname, ddname*; **ENTER**
"RETRY" OR "CANCEL"

- DFSMSrmm was unable to validate a mounted volume, or record an open or close of the volume *volser* because the DFSMSrmm subsystem was not active.
- No tape activity is allowed while DFSMSrmm is inactive. Start the DFSMSrmm subsystem, then enter RETRY to continue, or CANCEL to cancel the action.

System Error Messages

DFSMSrmm issues system error messages when errors occur, such as internal abends within DFSMSrmm, unsuccessful logging of changes to the DFSMSrmm control data set, unsuccessful journal processing, or I/O errors on the control data set. All DFSMSrmm error messages are written to the system log.

Notify your system programmer if any of these errors persist.

Following are system error messages for which you might want to set up handling procedures:

EDG0154I SHUTDOWN OF DFSMSRMM DELAYED BY ANOTHER ADDRESS SPACE

- Find the address space delaying shutdown by using the D GRS command to see which address space is holding the SYSZRMM SHUTDOWN resource.
- Reply to any outstanding WTORs for the address space holding the resource.

EDG2103D PERMANENT JOURNAL ERROR - REPLY "R" TO RETRY, "I" TO IGNORE, "D" TO DISABLE, OR "L" TO LOCK

- Look for a previous message with the EDG prefix that shows the error.
- Notify your system programmer.

EDG2104E JOURNAL FILE IS FULL - SCHEDULE CONTROL DATA SET BACKUP TO CLEAR IT.

- Manually start DFSMSrmm backup job to reset journal.
- Notify your system programmer.
- There is no reply for this message. This message is followed by message EDG2103D, which must be replied to.

EDG2107E JOURNAL THRESHOLD REACHED - JOURNAL IS
percentage_value% **FULL. tracks TRACKS (kilobytesK) AVAILABLE**

- The journal has reached the specified threshold value. If an auto-start procedure for backup is defined, then RMM starts it automatically. Otherwise follow your installation-defined backup procedure.
- Notify your system programmer.

EDG2108E JOURNAL IS *percentage_value%* **FULL. tracks TRACKS**
(kilobytesK) AVAILABLE

- This message is issued for every additional 5% full, or every 1% once over 90% full. If no backup procedure has already been started, then follow your installation-defined backup procedure.
- Notify your system programmer.

EDG2110I DFSMSrmm DETECTED A FAILED CONTROL DATA SET UPDATE

- DFSMSrmm is attempting to recover the DFSMSrmm control data set. Wait for message EDG2111I or EDG2115I. You do not need to reply to this message.

EDG2111I DFSMSrmm STARTING AUTOMATIC RECOVERY OF THE CONTROL DATA SET

- DFSMSrmm is attempting to recover the DFSMSrmm control data set using the journal. Wait for message EDG2112I or EDG2115I. You do not need to reply to this message.

EDG2112I DFSMSrmm AUTOMATIC RECOVERY OF CONTROL DATA SET SUCCESSFUL

- Recovery was successful and you need not reply to this message.

EDG2113I AUTOMATIC RECOVERY OF CONTROL DATA SET COMPLETED BY ANOTHER SYSTEM

- Recovery is in progress. You do not need to reply to this message.

EDG2114I AUTOMATIC RECOVERY OF CONTROL DATA SET HAS FAILED

- Recovery failed. Wait for message EDG2116A. You do not need to reply to this message.

EDG2115I RECOVERY OF CONTROL DATA SET IS REQUIRED

- Automatic recovery is not possible. Wait for message EDG2116A. You do not need to reply to this message.

EDG2116A DFSMSrmm QUIESCED - START CONTROL DATA SET RECOVERY PROCEDURE

- You can start recovery processing now by following your installation-defined recovery procedures.

EDG9115I I/O DISABLED FOR DFSMSrmm PROBLEM DETERMINATION OUTPUT DATA SET, REASON CODE *reason_code*

- DFSMSrmm issues this message when DFSMSrmm cannot write to the PDA trace output EDGPDOX or EDGPDOY data sets.
- Notify your system programmer.

EDG9116I RENAME ERROR SWAPPING DFSMSrmm PDA DATA SETS OLD DATA SET =*olddsn*, NEW DATA SET =*newdsn*, RETURN CODE *return_code* REASON CODE *reason_code*

- DFSMSrmm issues this message when the PDA trace output EDGPDOX or EDGPDOY data sets cannot be swapped.
- Notify your system programmer.

DFSMSrmm Installation Exit Messages

DFSMSrmm provides installation exits which allow your installation to perform special processing. Following are installation exit messages for which you might want to set up handling procedures:

EDG0300I INSTALLATION EXIT *exit_name* HAS FAILED - COMPLETION CODE S *sss* Uuuuu

- DFSMSrmm called an installation exit that abnormally ended. DFSMSrmm issues message EDG0303D prompting an operator response.

EDG0301I INSTALLATION EXIT *exit_name* PARAMETER LIST IS INCORRECT

- DFSMSrmm called an installation exit that has passed back a return code of 16, indicating that the parameter list passed to it is not acceptable. DFSMSrmm issues message EDG0303D prompting an operator response.

EDG0302I INSTALLATION EXIT *exit_name* IS NOW DISABLED

- DFSMSRmm disabled an installation exit. DFSMSRmm issues this message in response to the operator DISABLE reply to EDG0303D.

EDG0303D INSTALLATION EXIT *exit_name* HAS FAILED - REPLY "RETRY", "CANCEL", "DISABLE" OR "CONTINUE", *volser*, *jobname*, *stepname*

- When an error is detected during installation exit processing, DFSMSRmm issues WTOR EDG0303D.

You can reply:

- CANCEL to fail the current request, but process all others.
- DISABLE to disable the installation exit and continue with DFSMSRmm processing of the current request ignoring the failure of the installation exit. All future requests will be processed without use of the installation exit.
- CONTINUE to continue with DFSMSRmm processing of the current request ignoring the failure of the installation exit. All future requests will be processed using the installation exit.
- RETRY to retry the current request. Reply RETRY after refreshing the installation exit.
- If CONTINUE or DISABLE is the reply, a message is issued informing that DFSMSRmm information for the volume might be incomplete or incorrect.
- Processing of errors in the installation exit, across multiple requestors is serialized so that only one WTOR is issued at a time.

EDG0304I CONTINUING WITH REQUEST - INFORMATION RECORDED BY DFSMSRmm MAY BE INCOMPLETE FOR *volser*, *jobname*, *stepname*

- The operator replied DISABLE or CONTINUE to message EDG0303D. The current request continues but as a result the information recorded in the DFSMSRmm control data set might be incomplete or incorrect. You do not need to reply to this message.

EDG0305I INSTALLATION EXITS REFRESHED

- DFSMSRmm issues this message if the installation exits were successfully refreshed. You do not need to reply to this message.

EDG8121D VOLUME *req_volser* RACK *rack_number* LOCATION *loc_name* BIN *bin_number* HOME LOCATION *home* - ENTER VOLUME INTO LIBRARY *lib_name* AND REPLY "RETRY", OTHERWISE REPLY "CANCEL" OR "CONTINUE"

- DFSMSRmm issues this message to prompt the operator to enter a volume into a system-managed library.

Initializing and Erasing Tape Volumes Manually

Use the LABEL procedure to run the EDGINERS utility so that you can process initialize and erase requests manually. With manual processing of EDGINERS, you can initialize and erase any specific tape volume, whether it is already defined to DFSMSRmm or not. Once you initialize a volume, it is defined to DFSMSRmm.

The LABEL procedure is supplied as EDGLABEL in the DFSMSRmm SAMPLIB data set. You must have a tape drive on which you can initialize tape volumes.

Tapes can also be relabeled by defining new resource profiles in the FACILITY class. If the user has access to the security resource, DFSMSRmm allows the tape volume VOL1 label to be created or destroyed at time of use, regardless of the volume's status.

The tape label version for ISO/ANSI version tape labels can be specified in the EDGINERS EXEC JCL. If no tape label is specified, DFSMSrmm uses the version number in the parmlib DEVSUPxx member as the default tape label version when initializing ISO/ANSI tapes. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on labeling tapes using the EDGINERS utility or by defining resource profiles.

Using the LABEL Procedure to Request EDGINERS Processing

To use the LABEL procedure, use the command in Figure 133 where:

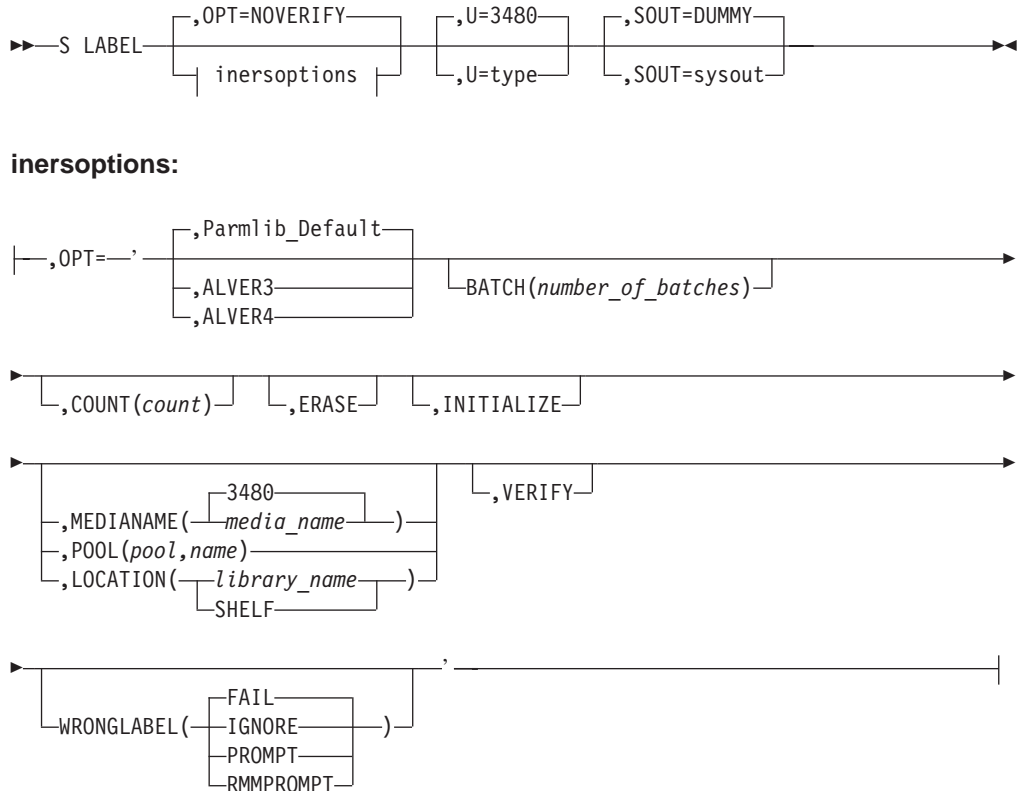


Figure 133. LABEL Command Syntax Diagram

OPT=NOVERIFY

This is the default value. NOVERIFY means that no additional verification is needed. If VERIFY is used, DFSMSrmm prompts you to remount each volume that has been erased or labeled. The volumes are requested in reverse order and the volume labels are read to ensure there are no label mismatches or other errors. Respond at the console to WTORs issued by EDGINERS.

OPT=inersoptions

You can use any valid combination of EDGINERS EXEC parameters as a substitute for *inersoptions*. Separate each parameter with a comma. If you use any parameters, other than VERIFY or NOVERIFY, you are requesting an automatic run of EDGINERS. During an automatic run, EDGINERS uses the DFSMSrmm control data set for input and not the console.

ALVER3

Use ALVER3 to initialize tape volumes with ISO/ANSI version 3 VOL1 and

HDR1 labels. If you do not specify ALVER3 or ALVER4, DFSMSrmm uses the default label value from the parmlib DEVSUPxx member.

ALVER4

Use ALVER4 to initialize tape volumes with ISO/ANSI version 4 VOL1 and HDR1 labels. If you do not specify ALVER3 or ALVER4, DFSMSrmm uses the default label value from the parmlib DEVSUPxx member.

BATCH(*number_of_batches*)

Use BATCH to specify the number of batches of volumes to be processed in a single run of EDGINERS in automatic mode. Use the COUNT parameter to specify the batch size. Batch size is the number of volumes that will be initialized or erased before those volumes are verified. After verification is completed for one batch, EDGINERS starts again for the next batch.

If no verification is requested, the number of volumes processed is the BATCH value or its default, multiplied by the value of COUNT or its default. However, DFSMSrmm does not batch the processing of these volumes.

The default for BATCH is BATCH(1). If all volumes in a location or pool with actions pending are to be processed, specify BATCH(0). DFSMSrmm treats BATCH(0) as BATCH(X'FFFFFFFF'), the upper limit for the number of batches that DFSMSrmm can process.

COUNT(*count*)

Use COUNT to specify the number of volumes to be selected for erasure or initialization by DFSMSrmm. The maximum value you can use is 99. If automatic processing is in effect but COUNT is omitted, then the default value is 10.

ERASE

Use ERASE to request that DFSMSrmm select volumes that have the erase action pending. If automatic processing is in effect but ERASE is not specified, then DFSMSrmm will only select volumes with the initialize action pending.

INITIALIZE

Use INITIALIZE to request that DFSMSrmm select volumes that have the initialize action pending. If automatic processing is in effect but neither INITIALIZE nor ERASE are specified then INITIALIZE is the default. You can also use INITIALISE for INITIALIZE.

LOCATION(*library_name*)

Specifies a subset of volumes for automatic processing.

The library name must be one that is defined as a system-managed tape library on the running system.

There is no default value.

MEDIANAME(*media_name*)

Specifies a subset of volumes for automatic processing. The default value is 3480, but is not used to set a default for the SYSIN INIT and ERASE commands MEDIANAME operand. The *media_name* must be one that matches to volumes defined to DFSMSrmm.

You can use only one of the operands, MEDIANAME, POOL, and LOCATION. If none of these operands are used, all volumes defined as 3480 media name will be selected if they have the required action pending.

For automatic processing VERIFY is the default. For manual processing NOVERIFY is the default.

Parmlib_Default

DFSMSrmm uses the default label value from the parmlib DEVSUPxx member if you do not specify ALVER3 or ALVER4.

POOL(pool_name)

Specifies a subset of volumes for automatic processing. Use a value one to five alphanumeric characters long plus an *. The pool must be one that is defined to DFSMSrmm on the running system.

There is no default value.

VERIFY

Use VERIFY to request that DFSMSrmm ask the operator to remount each volume that has been successfully erased or labeled. The volumes are requested in reverse order, and the volume labels are read to ensure that no operator errors have occurred, for example, a mismatch between the internal label and the external label.

WRONGLABEL

When DFSMSrmm detects that a wrong volume is mounted, it checks to see if the volume is defined to DFSMSrmm.

Use WRONGLABEL to specify the processing DFSMSrmm performs when the volume is mounted. WRONGLABEL can be used when running EDGINERS in both automatic and manual mode.

FAIL

DFSMSrmm does not prompt the operator to accept a mounted volume that does not match the requested volume. The mount request is rejected, the volume demounted, and DFSMSrmm issues message EDG6661E or EDG6662E.

IGNORE

When the wrong volume is mounted, no operator involvement is required and processing proceeds after issuing a message to log the label that was detected. Either message EDG6661E or EDG6662E is issued to log the relabeling. This is an extremely dangerous option and should only be used with great caution. ANY volume can be relabeled as long as the requested volume has the INIT action or is not defined to DFSMSrmm. Use of this option requires CONTROL access to RACF FACILITY class resource STGADMIN.EDG.INERS.WRONGLABEL.

PROMPT

When an incorrect volume label is detected by EDGINERS for the mounted volume, the operator is always prompted to confirm the processing to be performed. DFSMSrmm issues either message EDG6661E or EDG6662E, followed by message EDG6663D. Processing continues according to the response to message EDG6663D. This option should be used with caution as ANY volume can be relabeled as long as the requested volume is either known to DFSMSrmm and has

the INIT action, or is not known to DFSMSrmm. No additional authorization is required, other than that required for running EDGINERS.

RMM PROMPT

When the volume serial number of the mounted volume does not match the volser of the requested volume and the mounted volume is defined to DFSMSrmm, DFSMSrmm issues message EDG6663D to prompt the operator to confirm the processing to be performed. If the volser of the tape is not known to DFSMSrmm, initialization continues as if the tape had no label. If the volume is known to DFSMSrmm, DFSMSrmm issues messages EDG6662E and EDG6663D for the PROMPT option; otherwise EDG6661E is issued to log the relabeling. Use the PROMPT option only if all your volumes are known to DFSMSrmm; otherwise, caution is required as for use of PROMPT. UPDATE access to RACF FACILITY class resource STGADMIN.EDG.INERS.WRONGLABEL is required to use this option.

U=3480

This is the default value. Use the U keyword to identify either the device number or device type to be used for tape initialization or erase processing.

U=type

Use the U keyword to identify either the device number or device type to be used for tape initialization or erase processing. Your installation can define any *type* that is meaningful.

SOUT=DUMMY

This is the default value. It is used to ensure that the EDGINERS SYSPRINT file is not produced.

SOUT=sysout

Use SOUT to provide a valid DD statement keyword combination that is used for the EDGINERS SYSPRINT file. For example, by using SOUT='SYSOUT=A', the SYSPRINT file is produced in sysout class A. If this is a class that is available for printing, you can print the report and review any messages that DFSMSrmm has produced.

Tape mounts are issued for each volume you request to be initialized.

Replying to LABEL Procedure Messages

DFSMSrmm issues the following messages requiring your reply or action. See Figure 134 on page 215 for the command syntax for the INIT and ERASE commands used to reply to LABEL procedure messages.

EDG6626A SPECIFY VOLUME "INIT" OR "ERASE" COMMAND OR "END"

- Respond with either a volume INIT or ERASE command.
- You might want to modify DFSMSrmm to handle this automatically.

EDG6627A M *drive_number* V(*volser*) R(*rack_number*) TO BE *erased_or_labelled_or_verified*, *label_type*

- Mount the volume, write enabled. You do not need to reply to this message. DFSMSrmm returns a null *rack_number* when the volume is not defined to DFSMSrmm.
- If the volume cannot be mounted, reply 'S' to skip the volume.

EDG6628A *drive_number* REPLY WITH RACK NUMBER OR VOLUME SERIAL NUMBER FOR NL VOLUME

- The volume does not have a volume label, so DFSMSrmm cannot verify that the correct volume is mounted. Check that the correct volume is mounted.
- Reply with the volume serial number or rack number.

EDG6642I VOLUME *volser* LABELLED SUCCESSFULLY

- The volume was successfully labeled. Return it to its shelf location.

EDG6621E VOLUME *volser* INITIALIZATION FAILED.

- The volume was not labeled. Give it to the tape librarian for action.

EDG6620I VOLUME *volser* INITIALIZATION AND VERIFICATION SUCCESSFUL - RETURN TO RACK NUMBER *rack_number*

- The volume was internally labeled. Return it to the shelf location identified in the message by rack number.

EDG6622I VOLUME *volser(oldvolser)* INITIALIZATION SUCCESSFUL - RETURN TO RACK NUMBER *rack_number*

- The volume was internally labeled. Return it to the shelf location identified in the message by rack number.

EDG6623I VOLUME *volser(oldvolser)* ERASE, INITIALIZATION AND VERIFICATION SUCCESSFUL - RETURN TO RACK NUMBER *rack_number*

- The volume was internally labeled. Return it to the shelf location identified in the message by rack number.

EDG6625I VOLUME *volser(oldvolser)* ERASE, INITIALIZATION SUCCESSFUL - RETURN TO RACK NUMBER *rack_number*

- The volume was internally labeled. Return it to the shelf location identified in the message by rack number.

EDG6619I NO *unit_type* VOLUMES WERE FOUND WITH THE *initialize_or_erase* ACTION PENDING

- No volumes of the specified unit or type of media were found that needed to be initialized or erased.
- You do not need to respond to this message.

EDG6631I UTILITY *utility* COMPLETED WITH RETURN CODE *return_code*

- A tape label, erase, or verification job completed with a return code.
- You do not need to reply to this message.

EDG6658I VOLUME *volser(oldvolser)* IS MISSING SERVO TRACKS. PLEASE RETURN CARTRIDGE TO YOUR SUPPLIER TO BE REFORMATTED

- DFSMSrmm was unable to read the existing volume label from the volume because the volume servo information is not formatted.
- You do not need to respond to this message.

EDG6661E INCORRECT VOLUME MOUNTED ON DEVICE *device* - REQUESTED VOLUME *volser(oldvolser)* MOUNTED VOLUME *mounted_volume*

- The user tried to relabel a volume, *oldvolser*, to *volser*, but the wrong volume, *mounted_volume*, was mounted instead. *oldvolser* is not known to DFSMSrmm. DFSMSrmm detected that an incorrect volume was mounted. EDGINERS continues processing as required by the EXEC statement PARM WRONGLABEL specification.

When WRONGLABEL is	DFSMSrmm
FAIL	Does not prompt the operator for a reply and rejects the volume mount request.
IGNORE	Does not prompt the operator for a reply and relabels the volume if DFSMSrmm is requested to initialize the volume or if the volume is not managed by DFSMSrmm. Use of IGNORE requires CONTROL access to the RACF FACILITY class resource STGADMIN.EDG.INERS.WRONGLABEL.
PROMPT	Issues message EDG6663D to prompt the operator to identify the processing that should be performed. No additional authorization is needed to use PROMPT.

- You do not need to respond to this message.

EDG6662E INCORRECT VOLUME MOUNTED ON DEVICE *device* - REQUESTED VOLUME *volser(oldvolser)* MOUNTED VOLUME *mounted_volume* IS DFSMSrmm MANAGED

- The user tried to relabel a volume, *oldvolser*, to *volser*, but the wrong volume, *mounted_volume*, was mounted instead. *oldvolser* is defined to DFSMSrmm. EDGINERS continues processing as required by the EXEC statement PARM WRONGLABEL specification.
- You do not need to respond to this message.

EDG6663D REPLY "R" TO RETRY OR "F" TO FAIL THE REQUEST, OR "A" TO ACCEPT THE MOUNTED VOLUME

- During EDGINERS processing, DFSMSrmm detected an incorrect volume serial number on a mounted volume. The EXEC statement PARM WRONGLABEL parameter has been specified and DFSMSrmm prompts the operator to reply. DFSMSrmm issues this message after issuing either message EDG6661E or EDG6662E.
- Reply as described in Table 22.

Table 22. Operator Responses to Message EDG6663D

To	Reply	And the result is
Accept the mounted volume	A	DFSMSrmm relabels the volume to the new volume serial number.
Fail the request	F	DFSMSrmm unloads the mounted volume and the request fails.
Retry to request	R	DFSMSrmm unloads the mounted volume and reissues the mount request message EDG6627A.

EDG6672I A LABELVERSION VALUE HAS BEEN SPECIFIED FOR A LABEL TYPE OTHER THAN AL

- You specified the LABELVERSION operand for standard label or no label tape. You should only specify a value for ISO/ANSI label tapes.
- Reply to message EDG6626A with the INIT or ERASE command, or END to end the EDGINERS utility.

EDG6673I VALIDITY CHECK FAILED FOR *field* VALUE SPECIFIED WITH LABEL *type* AND VERSION *number* FOR VOLUME *volser*

- Invalid characters were entered in either the ACCESS(*code*) or OWNERTEXT(*text*) parameters in the INIT/ERASE statement.
- Reply with valid values.

INIT and ERASE Request Syntax

Figure 134 describes the syntax for replying to prompts for an INIT or ERASE request:

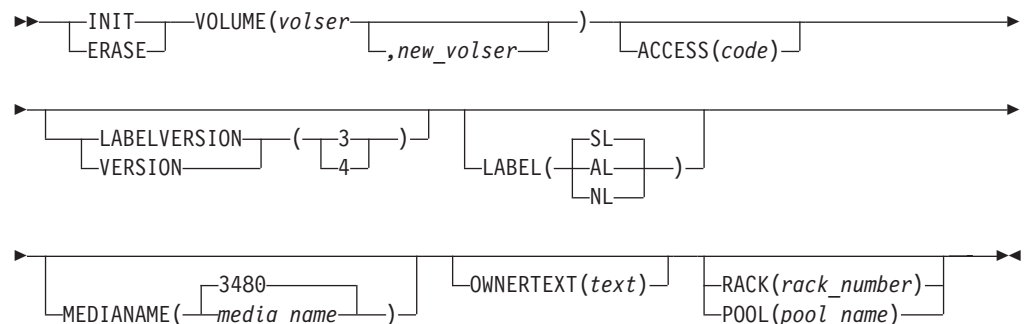


Figure 134. INIT and ERASE Request Syntax Diagram

INIT

Specify INIT to initialize a volume.

You must specify either INIT or ERASE.

ERASE

Specify ERASE to security erase a volume and write a new label on it.

You must specify either ERASE or INIT.

VOLUME(volser,new_volser)

volser specifies the volume serial number of the volume to be initialized or erased. *Volser* is required. If you are adding volumes with a volume serial number less than six characters, you must supply a rack number or a pool, otherwise DFSMSrmm issues an error message.

If the volume is already defined in the DFSMSrmm control data set, DFSMSrmm ensures that the requested action is pending for the volume. If this action is not pending, DFSMSrmm fails the request.

If the volume mounted is already labeled, DFSMSrmm reads the label to ensure that the volume serial number matches the one you specify. If the volume mounted does not have a recognizable volume label but contains data (no label tapes or nonstandard label tapes), DFSMSrmm issues a WTOR. The operator must reply to this message before DFSMSrmm can initialize or erase the volume.

If the volume is not defined in the DFSMSrmm control data set and you do not specify a new volume serial number, DFSMSrmm adds the volume to the control data set.

new_volser specifies a new volume serial number. Use it if you want to label a volume with a new volume serial number. If this new volume is already defined in the DFSMSrmm control data set, DFSMSrmm fails the request.

DFSMSrmm adds information about the new volume to the DFSMSrmm control data set, using information recorded for the volume you are replacing, and then deletes information about the volume you are replacing.

ACCESS(*code*)

Specifies the ISO/ANSI volume accessibility code. A valid code is in uppercase, alphabetic characters. You must specify LABEL(AL) if you specify an accessibility code.

You must modify the volume access installation exit routine in MVS/ESA to allow subsequent use of the volume if you specify ACCESS.

The default is blank, allowing unlimited access to the volume.

LABEL(AL/NL/SL)

Use LABEL to specify the type of label to be written on the volume:

AL

Specifies an ISO/ANSI Label.

NL

Specifies no label.

SL

Specifies a standard label.

If you do not specify the label type and the volume is already defined in DFSMSrmm, DFSMSrmm uses the label type defined in the DFSMSrmm control data set.

If you do not specify the label type and the volume is not already defined in the control data set, DFSMSrmm uses IBM standard label (SL) as the default.

LABELVERSION(3|4)

Specifies the ISO/ANSI volume label version for AL tape labels.

Valid values are 3 or 4 only. The default is the value specified in the parameter field of the EXEC JCL statement. If nothing is specified in the EXEC statement, the default is taken from the DEVSUPxx member in parmlib.

Using this parameter is equivalent to using the CHANGEVOLUME *volser* REQUIREDLABELVERSION subcommand, in that it sets the required label version for ISO/ANSI output tapes in the control data set volume record for this volume.

You must specify LABEL(AL) if you specify LABELVERSION.

MEDIANAME(*media_name*)

Specifies the volume's media name.

If the volume is already defined in the DFSMSrmm control data set, DFSMSrmm compares the value you specify to the media name defined in the control data set, and fails the request if they do not match.

If the volume is not defined in the DFSMSrmm control data set, DFSMSrmm uses the value you specify when it adds information about the volume to the control data set. If the volume is not already defined in the control data set and you do not specify a media name, DFSMSrmm uses the media name, 3480.

The default is 3480.

OWNERTEXT(*text*)

Specifies the owner's name or similar identification. *text* is fourteen characters.

Enclose in single quotes if it includes blanks or special characters. The text must be 10 bytes for SL, 14 bytes for AL.

The information is specified as character constants, and can be up to 10 bytes long for EBCDIC and BCDIC volume labels, or up to 14 bytes long for volume labels written in ASCII.

POOL(*pool_name*)

Specifies a pool ID for a pool to which you want to assign the volume. If the volume is not defined to DFSMSrmm, DFSMSrmm selects an available rack number for the volume in the pool you specify. If the volume is already defined in the DFSMSrmm control data set, DFSMSrmm changes the volume's rack number to move the volume.

If you do not supply a pool ID or a rack number and the volume is already defined in the DFSMSrmm control data set, DFSMSrmm uses the volume's existing rack number. If the volume is not defined in the control data set and you do not supply a pool ID or a rack number, DFSMSrmm assigns the volume a rack number matching its volume serial number.

RACK(*rack_number*)

Specifies a shelf location for the volume. If the volume is already defined in the DFSMSrmm control data set, DFSMSrmm compares the value you specify, and fails the request if it does not match the value in the control data set.

If you do not supply a pool ID or a rack number and the volume is already defined in the control data set, DFSMSrmm uses the volume's existing rack number. If the volume is not defined in the DFSMSrmm control data set and you do not supply a pool ID or a rack number, DFSMSrmm assigns the volume a rack number matching its volume serial number.

For example, if you want to initialize new volume ABC123 with an ISO/ANSI label and assign it to a shelf location in pool AB*, use the following command format to issue your operator reply:

```
R 21,INIT VOLUME(ABC123) LABEL(AL) POOL(AB*)
```

Processing Sticky Labels

If your installation has implemented disposition control, you can request that DFSMSrmm produce labels. Labels can be produced using a WTO on route code 13 or using the OUTPUT JCL statement to send labels to a spool file or a printer. You control the method DFSMSrmm uses by specifying an OUTPUT JCL statement in the DFSMSrmm started procedure. The name on the OUTPUT JCL statement must match the name specified for the DD name of the disposition control file.

If you use the OUTPUT JCL statement method, DFSMSrmm dynamically allocates a sysout file for each label by using the DISPDDNAME OUTPUT JCL statement. You use the attributes of the OUTPUT statement to define how the label output is to be printed. For example, you can route the output to another system, specify a special forms type or use any of the OUTPUT statement keywords.

If you do not provide an OUTPUT JCL statement in the DFSMSrmm started procedure, you must configure a console to accept WTO messages on route code 13 so that the labels can be printed.

Labels

The default label is 10 rowdata characters that are supported in a labels with a maximum of 80 characters per row. The default LRECL is 80. The maximum number of data characters supported in a label is 2000 characters. DFSMSrmm provides two default label styles for your use. You can change these label styles by using the DFSMSrmm EDGUX100 installation exit. Figure 135 shows the default label for cartridges which consists of eight data rows. Cartridge labels are identified by media type other than *, and a density of either *, IDRC, or 3480. Figure 136 shows the default label for all other types of volumes which consists of seven data rows and 2 or 3 rows for spacing the labels.

```
|...+...1....+...2....+...3....+...4....+...5....+...6....+...7.
dsname_____
userdata_____

jobname_      crdate____
              expdt_____
dens comp lrecl blksize recf

volser seqn lab      devc
```

Figure 135. Default Label Format for a Tape Cartridge

```
|...+...1....+...2....+...3....+...4....+...5....+...6....+...7.
dsname_____
userdata_____
              jobname_      crdate____
              expdt_____
dens comp lrecl blksize recf expdt_____

              volser seqn lab      devc
```

Figure 136. Default Label Format for a Round Tape

The values for the variables that are shown in Figure 135 and Figure 136 are:

dsname

The data set name of the file being processed. 1 to 44 characters.

userdata

The user data specified via message text in the disposition control file. 0 to 69 characters.

jobname

The current job name. 1 to 8 characters.

crdate

The data set create date. 1 to 10 characters in the date format that is specified by the DATEFORM parmlib option.

expdate

The data set expiration date. 1 to 10 characters in the date format specified by the DATEFORM parmlib option.

dens

The recording density of the volume. 1 to 4 characters.

comp

Indication of compaction of data on the volume. 4 characters.

| *lrecl*
| The logical record length of the data. 1 to 5 characters.
|
| *blksiz*
| The block size of the data. 1 to 5 characters.
|
| *recf*
| The record format. 1 to 4 characters.
|
| *volser*
| The volume serial number. 1 to 6 characters.
|
| *seqn*
| The volume sequence number. 1 to 4 characters.
|
| *lab*
| The volume label type. 1 to 3 characters.
|
| *devc*
| The number of the drive on which the file is processed. 4 characters.

Chapter 12. Using RMM TSO Subcommands

Use the RMM TSO command and a set of subcommands to request DFSMSrmm functions. This chapter describes each subcommand in detail. The commands are organized in alphabetical order and are illustrated using syntax diagrams.

Issuing the RMM TSO Command and Subcommands

Using the RMM TSO command and subcommands is an alternative to using the RMMISPF dialog. You can issue the RMM TSO command and subcommands from within the DFSMSrmm ISPF dialog or outside the dialog, in the foreground or by submitting a batch job,

You use TSO subcommands in two ways. For example, you can issue a request to add data set information by either:

```
RMM
ADDDATASET
:
END
```

or

```
RMM ADDDATASET ...
```

Always use the RMM TSO command, RMM, before entering subcommands and their operands. Once you have specified the RMM TSO command, RMM, you can continue entering subcommands. When you are ready to stop entering subcommands, specify:

```
END
```

You can issue the subcommands in full or abbreviated form. Table 23 lists the subcommands and their abbreviations.

Table 23. RMM TSO Subcommands

Group	Subcommand	Abbrev	Function
Add	ADDBIN	AB	Add bin number information
	ADDDATASET	AD	Add data set information
	ADDOWNER	AO	Add owner information
	ADDPRODUCT	AP	Add software product information
	ADDRACK	AR	Add shelf location information
	ADDVRS	AS	Add a vital record specification
	ADDVOLUME	AV	Add volume information
Change	CHANGEDATASET	CD	Change data set information
	CHANGEOWNER	CO	Change owner information
	CHANGEPRODUCT	CP	Change software product information
	CHANGEVOLUME	CV	Change volume information
Delete	DELETEBIN	DB	Delete bin number information
	DELETEDATASET	DD	Delete data set information
	DELETEOWNER	DO	Delete owner information
	DELETEPRODUCT	DP	Delete software product information
	DELETERACK	DR	Delete shelf location information
	DELETEVRS	DS	Delete a vital record specification
	DELETEVOLUME	DV	Release a volume and delete volume information

Table 23. RMM TSO Subcommands (continued)

Group	Subcommand	Abbrev	Function
Get	GETVOLUME	GV	Request or assign a volume
List	LISTBIN	LB	Display bin number information
	LISTCONTROL	LC	Display parmlib options and control information
	LISTDATASET	LD	Display data set information
	LISTOWNER	LO	Display owner information
	LISTPRODUCT	LP	Display software product information
	LISTRACK	LR	Display shelf location information
	LISTVRS	LS	Display vital record specification information
	LISTVOLUME	LV	Display volume information
Search	SEARCHBIN	SB	Create a list of bin numbers
	SEARCHDATASET	SD	Create a list of data sets
	SEARCHPRODUCT	SP	Create a list of software products
	SEARCHRACK	SR	Create a list of rack numbers
	SEARCHVRS	SS	Create a list of vital record specifications
	SEARCHVOLUME	SV	Create a list of volumes

Requesting Help for RMM TSO Subcommands

To request online help for TSO subcommands, enter:

```
TSO HELP RMM
```

To request help for a particular subcommand, enter RMM with a command abbreviation. For example, to view help for the ADDRACK command, enter:

```
TSO HELP RMMAR
```

Within the DFSMSrmm subcommand environment, you can simply enter:

```
RMM
HELP AR
:
END
```

Or you can use the command abbreviation, H, as follows:

```
RMM
H AR
:
END
```

Submitting a Batch Job

When you submit a batch job, it must execute the TSO TMP in the background. Input can be the RMM TSO command and subcommands, a CLIST or EXEC containing commands, or other valid TSO commands.

Following is an example of the JCL required to run a batch TSO TMP, with the DFSMSrmm TSO subcommands, RMM LISTCNTL and RMM LISTOWNER.

```
//TMP      EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
RMM
```

```
LISTCNTL ALL  
END  
RMM LISTOWNER owner  
/*
```

For more information about executing the TSO TMP in the background, refer to *OS/390 TSO/E User's Guide*.

ADDBIN Subcommand

ADDBIN: Adding a Bin Number in a Storage Location

The ADDBIN subcommand is an alias for the ADDRACK subcommand. See “ADDRACK: Adding a Shelf Location” on page 238 for the combined description of the ADDRACK and ADDBIN operands.

DFSMSrmm defines shelf space in storage locations as bin numbers. Use the ADDBIN subcommand as shown in Figure 137 on page 225 to define empty or available shelf locations in storage locations. The storage locations can be DFSMSrmm built-in storage locations or storage locations defined by your installation.

To add one or more bin numbers to an installation defined storage location, you must supply an initial bin number that identifies the shelf location, the installation defined location name, and a media name. You indicate the number of bin numbers you want to add, by using the COUNT operand. If you add more than one bin number at a time, the initial bin number should be long enough and low enough to accommodate the count value to be added to it without exceeding the numeric capabilities of the suffix. For example, if you supply an initial bin number of RA9992 and request that ten bin numbers be added, DFSMSrmm issues a warning message indicating that it cannot add all bin numbers.

To add bin numbers to a DFSMSrmm built-in storage location, use an * in place of a bin number and provide a built-in storage location name LOCAL, REMOTE, or DISTANT. You can provide the number of bins you want to add. DFSMSrmm automatically assigns the bin numbers.

You can use the LISTCONTROL subcommand with the CNTL operand to determine how many bin numbers are already defined and how many bin numbers are currently in use in the built-in storage locations. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333 for more information.

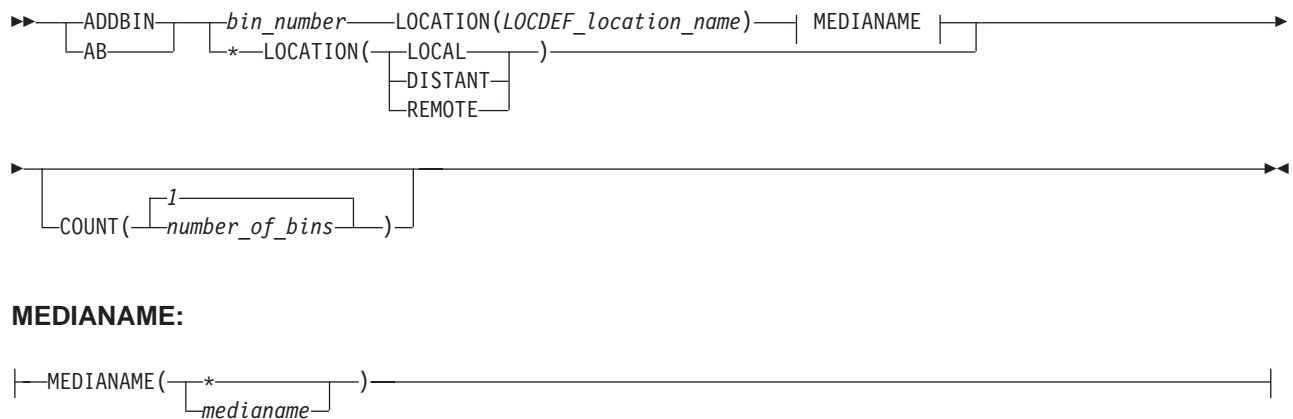
Example

Task: Add 5 empty shelf locations to the installation defined storage location, MYLOC, starting with bin number LN0002. Round media are accepted into this storage location.

Command:

```
RMM ADDBIN LN0002 LOCATION(MYLOC) COUNT(5) MEDIANAME(ROUND)
```


Syntax



MEDIANAME:

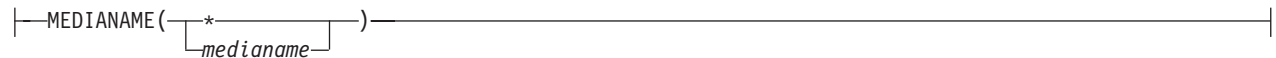


Figure 137. ADDBIN Syntax Diagram

Operands

bin_number, *

Use a bin number to define a shelf location in an installation defined storage location. A bin number in an installation defined storage location is six alphanumeric or national characters in any combination. You must also give a location name and media name.

Use an * to add bin numbers to a built-in storage location. If you use one of the built-in storage location names, LOCAL, DISTANT or REMOTE, DFSMSrmm determines the bin numbers that are used.

You must use either a bin number or an *, immediately following the ADDBIN subcommand.

COUNT(*number_of_bins*)

Specifies how many bin numbers to add to a storage location. The value is one to five numbers. The maximum allowable decimal value is 99999.

The default value is 1.

LOCATION(LOCAL,DISTANT,REMOTE,LOCDEF_location_name)

Specifies the storage location where you want to add shelf space. LOCAL, DISTANT, and REMOTE are DFSMSrmm built-in storage location names. You cannot use MEDIANAME with a built-in storage location name.

LOCDEF_location_name can be any name up to eight characters. It is the installation defined storage location name defined on LOCDEF in the current parmlib. To add bin numbers to an installation defined storage location, you provide the bin numbers to use. MEDIANAME must also be specified if you use a *LOCDEF_location_name*.

MEDIANAME(*medianame*,*)

Defines the media that can reside in a shelf location. *medianame* can be up to eight characters and must appear in the MEDIANAME value in the parmlib LOCDEF parameters for the LOCATION specified or DFSMSrmm rejects the request. If an * is specified, bins are allocated for use by volumes of any media name. Any other media name specifies that only volumes of that media name can be allocated to the bin number.

ADDBIN Subcommand

Example

Task: Add ten empty bin numbers to the LOCAL built-in storage location.

Command:

```
RMM ADDBIN * LOCATION(LOCAL) COUNT(10)
```

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDDATASET: Adding Data Set Information

Use the ADDDATASET subcommand as described in Figure 138 on page 228 to manually define a data set on a volume to DFSMSrmm. The volume on which the data set resides must have either master or user status.

When you define a data set to DFSMSrmm, you must supply the data set name and the serial number of the volume where it resides. Use the FILESEQ operand when you add information about a data set that is not the first data set on the volume.

To retain a data set by job name, create a data set vital record specification by specifying a job name with the ADDVRS subcommand. Then use the JOBNAME operand with the ADDDATASET subcommand to add the job name that created the data set to the data set information. During vital records processing, if no job name is defined for the data set, DFSMSrmm uses the data set name to retain a data set.

Before You Use This Subcommand

Data sets preceding the data set you want to add must already be defined to DFSMSrmm, or your request fails. You can use this subcommand to add information about all preceding data sets on the volume to DFSMSrmm if DFSMSrmm has no record of them.

Information about the volume where the data set resides must be defined to DFSMSrmm before you can add information about the data set. Use the ADDVOLUME subcommand to define volumes to DFSMSrmm. See “ADDVOLUME: Adding Volume Information” on page 241 for more information.

Example

Task: Add a data set named PREFIX.MYDATA.DATA that has a record format of fixed block with a record length of 80 bytes that resides on volume 8E1U01.

Command:

```
RMM ADDDATASET 'PREFIX.MYDATA.DATA' VOLUME(8E1U01) LRECL(80) RECFM(FB)
```

or, if prefix is your own TSO PROFILE PREFIX, you can enter:

```
RMM ADDDATASET MYDATA.DATA VOLUME(8E1U01) LRECL(80) RECFM(FB)
```

ADDDATASET Subcommand

Syntax



Notes:

1. Specify to add a data set to a volume where information was recorded by DFSMSrmm during O/C/EOV processing. You must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource to use the FORCE operand.

Figure 138. ADDDATASET Syntax Diagram

Operands

BLKCOUNT(number_of_blocks)

Specifies the number of data blocks used by the data set. The value corresponds to that recorded in the data set's End of File marker. The minimum allowable decimal value is 0; the maximum allowable decimal value is 9999999. DFSMSrmm uses BLKCOUNT, together with BLKSIZE, to calculate the approximate size of the data set when the data set is closed. If you do not use BLKCOUNT, DFSMSrmm cannot list the space used for the data set or volume when you request it.

BLKSIZE(block_size)

Specifies the block size of the data set. The minimum allowable decimal value is 0; the maximum allowable decimal value is 32760. DFSMSrmm uses BLKSIZE together with BLKCOUNT to calculate the approximate size of the data set when the data set is closed. If you do not use BLKSIZE, DFSMSrmm cannot report space use.

CRDATE/DATE(*create_date*)

Specifies the date when the data set was first written to tape. Supply the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 93001.
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm defaults to the 20th century.

The default is the date when you issue the ADDDATASET subcommand.

CRTIME/TIME(*create_time*)

Specifies the time the data set was first written to tape. The format is *hhmmss* where:

- *hh* is hours
- *mm* is minutes
- *ss* is seconds

For example, nine o'clock in the morning is 090000.

The default is the time when you issue the ADDDATASET subcommand.

data_set_name

Specifies the name of the data set being added. The name follows standard MVS naming conventions for data sets. The data set name must not include a member name. This operand is required and must immediately follow the ADDDATASET subcommand.

DEVNUM(*device_number*)

Specifies the device number of the drive on which the volume was mounted when DFSMSrmm recorded information about the data set. Use a three or four character hexadecimal number, using leading zeros if the number is less than four digits.

FILESEQ/SEQ(*1,physical_file_sequence_number*)

Specifies the relative position of the data set on the volume. The minimum allowable decimal value is 1. The maximum allowable decimal value is 9999. When you add a data set that is not the first data set on a volume, the preceding data sets on the volume must already be defined to DFSMSrmm.

The default value is 1.

FORCE

Specify to override the restriction that information that DFSMSrmm recorded during O/C/EOV processing cannot be changed. Using FORCE allows you to add a data set to a volume where DFSMSrmm recorded information during O/C/EOV processing. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATED access to STGADMIN.EDG.FORCE security resource.

JOBNAME(*create_jobname*)

Specifies the name of the job that created the data set. A job name is one to

ADDDATASET Subcommand

eight alphanumeric characters or \$, #, or @. Job name must start with an alphabetic character, \$, #, or @. You cannot use a generic jobname. Any jobname you use must be specific.

If you do not specify JOBNAME, DFSMSrmm uses the data set name only to retain the data set.

LABELNUMBER(*data_set_sequence_number*)

Specifies the data set sequence number you have to enter on the LABEL JCL parameter for allocating the specific data set without using the catalog entry. The minimum allowable decimal value is 0. The maximum allowable decimal value is 9999. The value you specify is not validated with the values known for the preceding or following data sets on the volume.

The default value is 0.

LRECL(*logical_record_length*)

Specifies the length, in bytes, of the largest logical record in the data set. The minimum allowable decimal value is 0; the maximum allowable decimal value is 32760.

The default value is 0.

READDATE(*last_read_date*)

Specifies when the data set was last read. Supply the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 93001.
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm defaults to the 20th century.

If a vital record specification indicates that DFSMSrmm retain a data set by last reference days and you did not enter a read or write date for the data set, DFSMSrmm uses the data set's creation date.

DFSMSrmm updates the details for the volume on which the data set resides if the last read date you use is more recent than the last read date recorded for the volume.

RECFM(*record_format*)

Specifies the format and characteristics of the records in the data set.

U Records are of undefined length

F Fixed-length records

FB
Blocked fixed-length records

FS
Fixed-length, standard records

FBS
Fixed-length records, written as standard blocks.

V Variable-length records

VB
Blocked, variable-length record

VS
Variable-length, spanned records

VBS
Variable-length records, possibly spanning more than one block

D Variable-length ASCII records

DB
Blocked variable-length ASCII records

DS
Variable-length ASCII spanned records

DBS
Variable-length ASCII blocked spanned records

You can also append either A or M to the fixed and variable formats.

A The record contains ASCII printer control characters

M The record contains machine code control characters

For example, you can use FBA or FBM to indicate that the records in the data set are blocked fixed-length records containing either ASCII printer or machine code control characters.

You can also append A to one of D, DB, DS, or DBS. For example, you can use DBA to indicate that the records in the data set are blocked variable-length ASCII records containing ASCII printer control characters.

SECLEVEL(*security_class*)

Specifies the security class of the data set. The value is one to eight characters and must be previously defined for your installation. If you do not use SECLEVEL, DFSMSrmm uses the data set name and your installation security class definitions to determine the SECLEVEL.

Use the LISTCONTROL subcommand to display your installation's security classes. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on using security class definitions.

SYSID(*SMF_system_ID*)

Specifies an ID for the system where the data set was created. This can be the system ID you use for DFSMSrmm supplied in EDGRMMxx parmli member, or it can be the SMF ID for your system if you have not given a DFSMSrmm system identifier. The value must be one to eight alphanumeric characters, \$, #, or @, or special characters.

VOLUME(*volume_serial*)

Specifies the serial number of the volume where the data set resides. The volume must have either master or user status; it cannot be a scratch volume. A volume serial is one to six alphanumeric characters, \$, #, or @, or special characters. This operand is required.

ADDDATASET Subcommand

WRITEDATE(*last_write_date*)

Specifies when the data set was last written to tape. Supply the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 93001.
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm defaults to the 20th century.

If a vital record specification indicates that DFSMSrmm retain a data set by last reference days and you did not enter a read or write date for the data set, DFSMSrmm uses the data set's creation date.

DFSMSrmm updates the details for the volume on which the data set resides if the last write date you use is more recent than the last write date recorded for the volume.

Return Codes

See "Chapter 13. DFSMSrmm Return and Reason Codes" on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDOWNER: Adding Owner Information

Use the ADDOWNER subcommand as shown in Figure 139 on page 234 to define an owner to DFSMSrmm. An owner can be an individual or a group defined by a RACF group name, or any other value you choose.

DFSMSrmm automatically creates an owner record if a user who is not defined to DFSMSrmm requests a job that writes to a volume managed by DFSMSrmm. DFSMSrmm uses the user ID that requested the job as a DFSMSrmm owner ID. To use DFSMSrmm automatic owner notification, you must manually add the user ID and node to be used as an electronic address.

You must supply an owner ID and a department name. The owner ID can be the owner's RACF user ID or any name you select to identify a single owner or owner group to DFSMSrmm. We suggest that you use a RACF user ID or RACF group name.

To use DFSMSrmm automatic owner notification, define an owner's electronic address by supplying the USER and NODE operands. DFSMSrmm uses this address to notify the owner when the owner's volumes are eligible for release. Notification must be used as one of the actions to be performed upon the volume's release, and the parmlib option, OPTION NOTIFY, must be set to Y. Use the ADDVOLUME or CHANGEVOLUME subcommands to set release actions for a volume. See "ADDVOLUME: Adding Volume Information" on page 241 or "CHANGEVOLUME: Changing Volume Information" on page 282 for more information. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on setting parmlib options.

Example

Task: Add details of the following new owner:

Table 24. New Owner Information

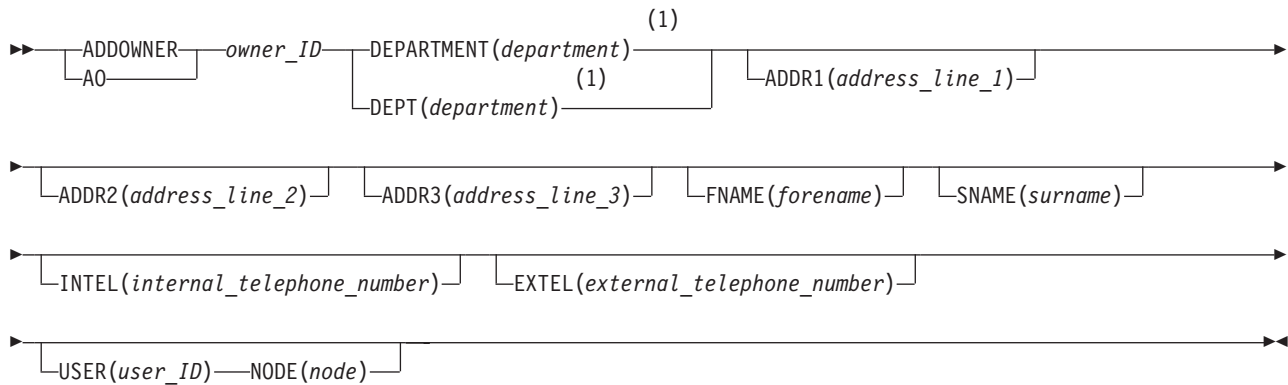
Owner's user ID	OWNERAS
Owner's department	Personnel
First line of owner's address	XYZ Company (UK)
Second line of owner's address	London
Owner's surname	Smith
Owner's initials	A B
Owner's internal telephone number	321 1234
Owner's electronic user ID	RANDSTER
Owner's electronic node ID	BUBVM30

Command:

```
RMM ADDOWNER OWNERAS DEPT('Personnel') ADDR1('XYZ Company (UK)') -
  ADDR2('London') SNAME('Smith') FNAME('A B') INTEL('321 1234') -
  USER(RANDSTER) NODE(BUBVM30)
```

ADDOWNER Subcommand

Syntax



Notes:

1. Cannot be all blank characters

Figure 139. ADDOWNER Syntax Diagram

Operands

ADDR1(address_line_1)

Specifies the first address line. An address line is one to forty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

ADDR2(address_line_2)

Specifies the second address line. An address line is one to forty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

ADDR3(address_line_3)

Specifies the third address line. An address line is one to forty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

DEPARTMENT/DEPT(department)

Specifies the owner's department name. A department name is one to forty characters and must not be all blanks. Enclose the department name in single quotes if it contains any special characters or blanks. A department name is required.

EXTEL(external_telephone_number)

Specifies the owner's external telephone number. An external telephone number is one to twenty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

FNAME(forename)

Specifies the owner's forename, or first name, initials, or title. A forename is one to twenty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

INTEL(internal_telephone_number)

Specifies the owner's internal telephone number. An internal telephone number is one to eight characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

ADDOWNER Subcommand

NODE(*node*)

Specifies the node name for electronic communication to the owner. A node ID is one to eight alphanumeric characters, \$, #, or @. If you use NODE, you must also use USER. The default is no electronic mail address.

owner_ID

Specifies an owner ID. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. We suggest that you use a RACF user ID or RACF group name. This operand is required and must immediately follow the ADDOWNER subcommand.

SNAME(*surname*)

Specifies the owner's surname, or last name. A surname is one to twenty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

USER(*user_ID*)

Specifies the owner's user ID to be used for electronic communication. A user ID is one to eight alphanumeric characters. If you specify USER, you must also use NODE. The default is no electronic mail address.

Return Codes

See "Chapter 13. DFSMSrmm Return and Reason Codes" on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDPRODUCT Subcommand

ADDPRODUCT: Adding Software Product Information

Use the ADDPRODUCT subcommand as shown in Figure 140 to define a software product to DFSMSrmm. You must enter the product number and name when adding a software product to DFSMSrmm.

After you have defined the software product to DFSMSrmm, use the ADDVOLUME subcommand to associate one or more volumes with the software product. The software product must be defined to DFSMSrmm before you associate one or more volumes with it, and the volumes must not be already defined to DFSMSrmm. Use the ADDVOLUME subcommand to define new volumes. See “ADDVOLUME: Adding Volume Information” on page 241 for more information.

If you are adding a newer release of the software product and the volume you use has the same volume serial number as a volume already defined to DFSMSrmm, you might want to consider disposing of the older release and deleting the volume information from DFSMSrmm. To keep the older release and avoid duplicate volume serial numbers, you must redefine the old software product volume before you can add the new software product volume. See “Redefining a Volume Already Defined to DFSMSrmm” on page 60 for more information on redefining volumes.

Use the CHANGEPRODUCT subcommand to change or add missing information about the software product. See “CHANGEPRODUCT: Changing Software Product Information” on page 280 for more information.

If your installation has set up the NOTIFY option, the designated product owner will be notified if a program product volume is added. The owner record must contain a valid user ID and node ID.

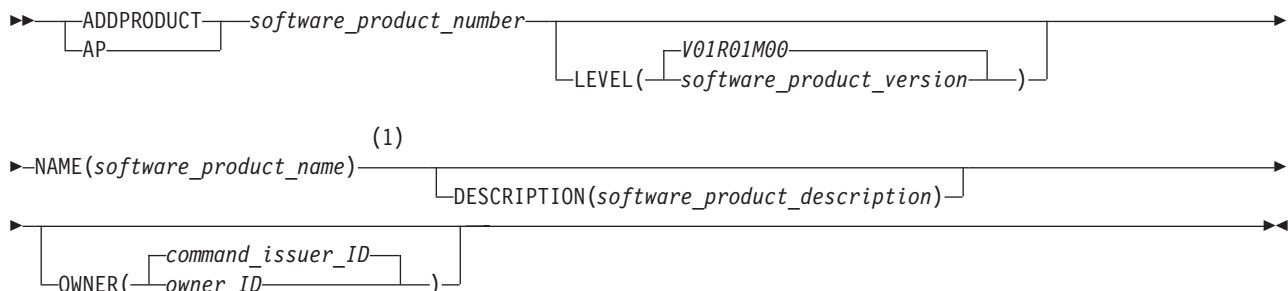
Example

Task: Add details of the new DFSMS/MVS software product with product number 5695-DF1, version 1.2.0.

Command:

```
RMM ADDPRODUCT '5695-DF1' NAME('DFSMS/MVS') LEVEL(V01R02M00)
```

Syntax



Notes:

1. Cannot be all blank characters

Figure 140. ADDPRODUCT Syntax Diagram

Operands

DESCRIPTION(*software_product_description*)

Specifies descriptive text about the software product. The descriptive text is one to thirty characters enclosed in single quotes if it contains any special characters or blanks.

LEVEL(*software_product_version*)

Specifies the software product's version. Supply the version in the form, VnnRnnMnn, indicating the version, release, and modification level. 'nn' is two numbers in the range 00 to 99.

The default value is V01R01M00, Version 1, Release 1, Modification 0.

NAME(*software_product_name*)

Specifies the software product's name. A software product name is one to thirty characters enclosed in single quotes if it contains any special characters or blanks. This operand is required and must not be all blanks.

You can use the value you specify for NAME with the SEARCHPRODUCT subcommand to request lists of software products defined to DFSMSrmm. See "SEARCHPRODUCT: Creating a List of Software Products" on page 369 for more information.

OWNER(*owner*)

Specifies the software product's designated owner. An owner ID is one to eight alphanumeric characters, \$, #, or @; normally a RACF user ID or RACF group name. The first character must not be a number. The default is the user ID of the command issuer.

software_product_number

Specifies the software product's number or ID. A software product number is one to eight characters enclosed in single quotes if it contains any special characters or blanks. This operand is required and must immediately follow the ADDPRODUCT subcommand.

Return Codes

See "Chapter 13. DFSMSrmm Return and Reason Codes" on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDRACK Subcommand

ADDRACK: Adding a Shelf Location

This section describes the combined description for the ADDRACK subcommand and its alias ADDBIN. See “ADDBIN: Adding a Bin Number in a Storage Location” on page 224 for information about using the RMM ADDRACK subcommand alias.

Use the ADDRACK subcommand as shown in Figure 141 on page 239 to define shelf locations in the removable media library and storage locations. DFSMSrmm defines shelf space in the removable media library as rack numbers and bin numbers in storage locations. Use the ADDRACK subcommand to define empty, or available, rack and bin numbers to DFSMSrmm.

You can add rack numbers to particular pools in your removable media library, as defined by your installation. Pools are groups of rack numbers with a common prefix. Use the LISTCONTROL subcommand with the VLPOOL operand to view the pool IDs defined for your installation, as well as information on individual pools. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on pooling strategies.

To add one or more rack numbers to the removable media library, you must supply an initial rack number. You can also indicate how many rack numbers you want to add. If you add more than one rack number at a time, your initial rack number must be long enough and low enough to accommodate the count value to be added to it without exceeding the numeric capabilities of the suffix. For example, if you supply an initial rack number of RA9992 and request that ten rack numbers be added, DFSMSrmm issues a warning message indicating that it cannot add all rack numbers.

To add rack numbers to be used for volumes in a specific system-managed library in the removable media library, add them to a pool in that library. For example, if you have a manual tape library dataser in which pool KD* resides, you can add rack numbers to that library by using a rack number with the prefix KD as the initial rack number on the ADDRACK subcommand.

To add bin numbers to a built-in storage location, use * as the bin number and provide a storage location name. You can also say how many bins you want to add. DFSMSrmm automatically assigns the bin numbers. To add bin numbers to an installation defined storage location, provide an initial bin number, a storage location name, and a media name. You can also say how many bins you want to add.

To obtain information about bin numbers in built-in storage locations, use the LISTCONTROL subcommand with the CNTL operand to determine how many bin numbers are already defined and how many bin numbers are currently in use as shown in “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333.

See “LISTBIN: Displaying Information about a Shelf Location” on page 331 to obtain information about individual bin numbers.

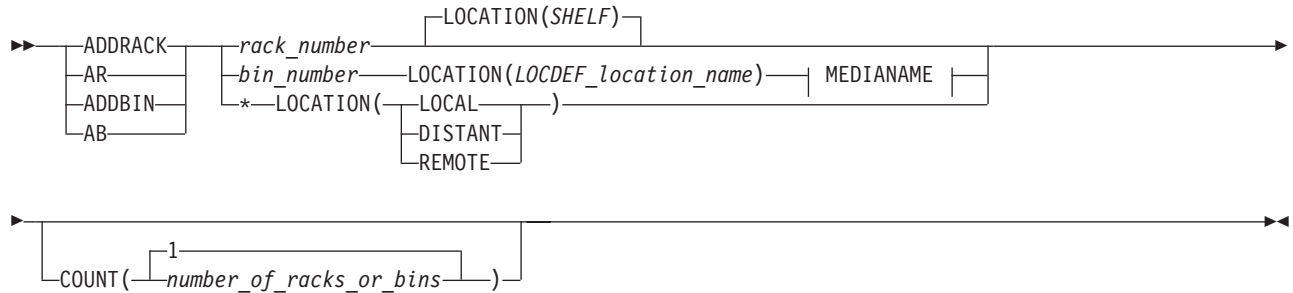
Example

Task: Add five empty racks to the removable media library, starting with rack number ABUB02.

Command:

RMM ADDRACK ABUB02 COUNT(5)

Syntax



MEDIANAME:

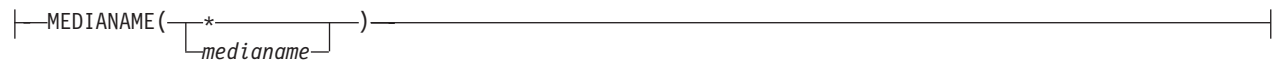


Figure 141. ADDRACK Syntax Diagram

Operands

*bin_number,**

Use a bin number to define a shelf location in an installation defined storage location. A bin number in an installation defined storage location is six alphanumeric or national characters in any combination. You must also give a location name and media name.

Use an * to add bin numbers to a built-in storage location. If you use one of the built-in storage location names, LOCAL, DISTANT or REMOTE, DFSMSrmm determines the bin numbers that are used.

You must use either a bin number or an *, immediately following the ADDBIN subcommand.

COUNT(*number_of_racks_or_bins*)

Specifies how many rack numbers to add to the removable media library or how many bin numbers to add to a storage location. The value is one to five numbers. The maximum allowable decimal value is 999999.

The default value is 1.

LOCATION(SHELF,DISTANT,LOCAL,REMOTE,LOCDEF_location_name)

Specifies the location where you want to add shelf space. Use SHELF to add shelf locations to your removable media library.

The DFSMSrmm built-in storage location names are LOCAL, DISTANT, and REMOTE. You cannot use MEDIANAME with DISTANT, LOCAL, or REMOTE when they are used as built-in storage location names.

LOCDEF_location_name can be a name up to eight characters long. It is the installation defined storage location name defined on LOCDEF in the current parmlib. To add bin numbers to an installation defined storage location, you provide the bin numbers that DFSMSrmm assigns. MEDIANAME must also be

ADDRACK Subcommand

specified. If you do not use the LOCATION operand, DFSMSrmm adds rack numbers to the removable media library.

MEDIANAME(*medianame*,*)

Defines the media that can reside in a shelf location. *medianame* can be up to eight characters and must appear in the MEDIANAME value in the parmlib LOCDEF parameters for the LOCATION specified or DFSMSrmm rejects the request. If an * is specified, bins are allocated for a volume of any media name. Any other media name specifies that only volumes of that media name can be allocated to the bin number.

MEDIANAME is not specified for adding rack numbers because the media name is obtained from the current VLPOOL definitions.

rack_number

Specifies a rack number to be added to the removable media library. This can be either a single rack number, or the initial rack number if you are adding more than one rack number. If you are adding multiple rack numbers, your initial rack number should contain numeric suffixes so that DFSMSrmm can automatically define each new rack number. A rack number is six alphanumeric or national characters in any combination.

You must use either a bin number, rack number or an *, immediately following the ADDRACK subcommand.

If you specify a *library_name* as the LOCATION value, the rack number must be the same as the volume serial number.

Example

Task: Add ten empty bin numbers to the LOCAL storage location using the alias ADDBIN.

Command:

```
RMM ADDBIN * LOCATION(LOCAL) COUNT(10)
```

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDVOLUME: Adding Volume Information

Use the ADDVOLUME subcommand as shown in Figure 142 on page 242 to add one or more volumes to DFSMSrmm.

You must supply a volume serial number and volume status. If you are adding more than one volume, you must supply an initial volume serial number and how many volumes you want to add (COUNT). If you use a pool prefix, DFSMSrmm assigns the volume the next available rack number in that pool or rack number range. If you do not use a pool prefix or a rack number, DFSMSrmm assigns the volume a rack number matching the volume serial number. If you are adding a volume with a volume serial number that is less than six characters, you must also include a rack number or a pool prefix.

Use the POOL or RACK operand to point to a shelf location where the volume should reside in the removable media library. You cannot specify POOL or RACK when you are adding a logical volume.

Use the MEDIANAME operand to indicate the name or type of media. If you add the volume to a pool, the media name you use must match the type of media defined for the pool or your ADDVOLUME request fails.

Use the LOCATION operand to indicate where the volume resides: either a shelf location in a non-system-managed tape library, or a system-managed tape library defined by your installation. This also sets the volume's home location, which is where you want a volume returned when it is no longer retained by a vital record specification.

Use the NUMBER, FEATCD, and LEVEL operands to associate the volume with a software product. The software product must be already defined to DFSMSrmm before you can associate it with a volume.

When you use ADDVOLUME STATUS(VOLCAT) DFSMSrmm uses information in the TCDB to update the DFSMSrmm control data set. The control data set information that can be updated includes: COMPACTION, CONTAINER, EXPDT, HOME, LOCATION, MEDIATYPE, READDATE, RECORDINGFORMAT, SPECIALATTRIBUTES, STATUS, STORAGEGROUP, TYPE, and WRITEDATE.

If you use any non-scratch operands when adding scratch volumes, DFSMSrmm ignores the non-scratch operands.

Example

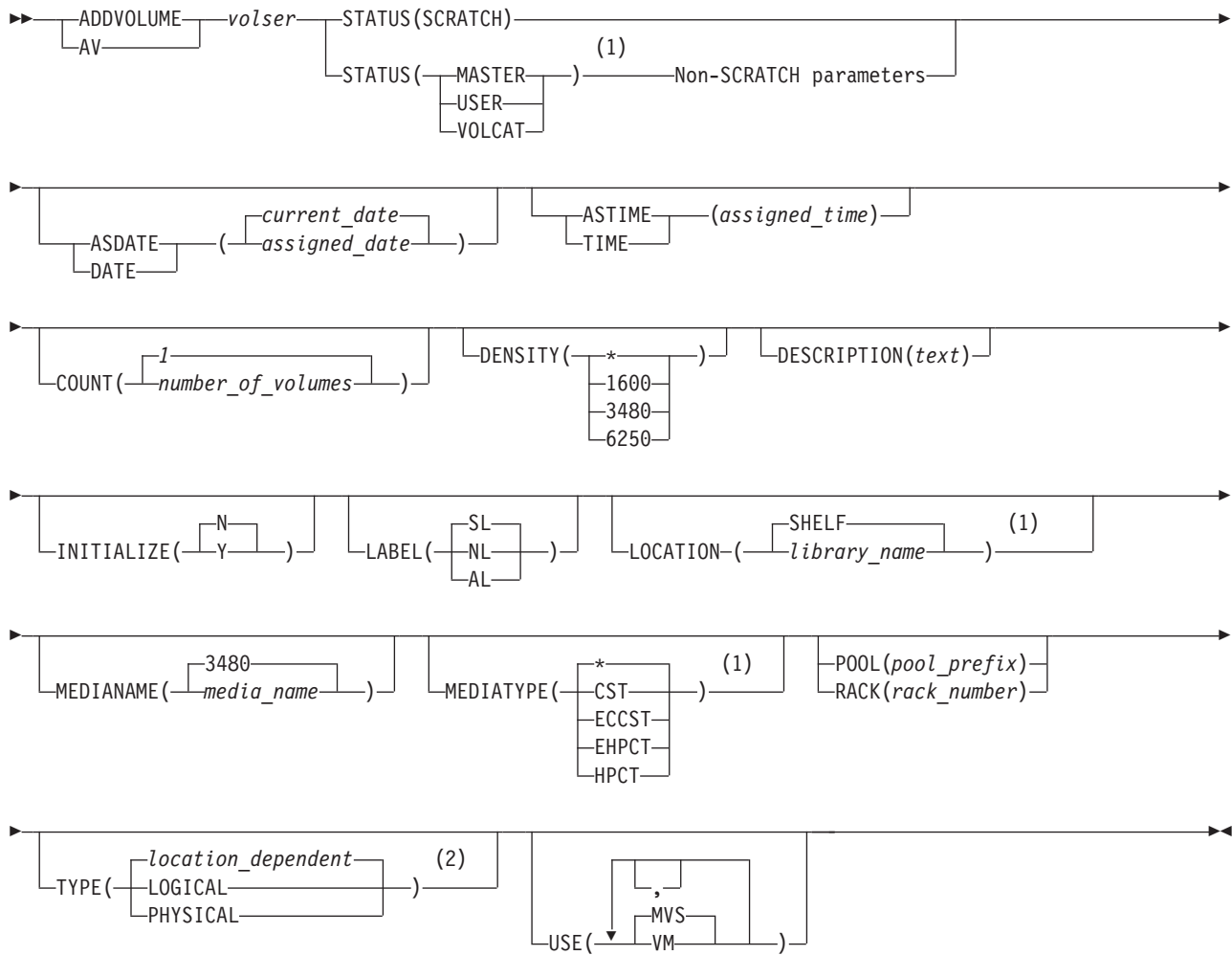
Task: Add 1000 new scratch volumes to the removable media library using a rack number that is the same as the volume serial number. The rack numbers have been already defined. The volumes must be labeled before they can be used, as indicated by INIT(Y).

Command:

```
RMM ADDVOLUME S00000 COUNT(1000) STATUS(SCRATCH) INIT(Y)
```

ADDVOLUME Subcommand

Syntax

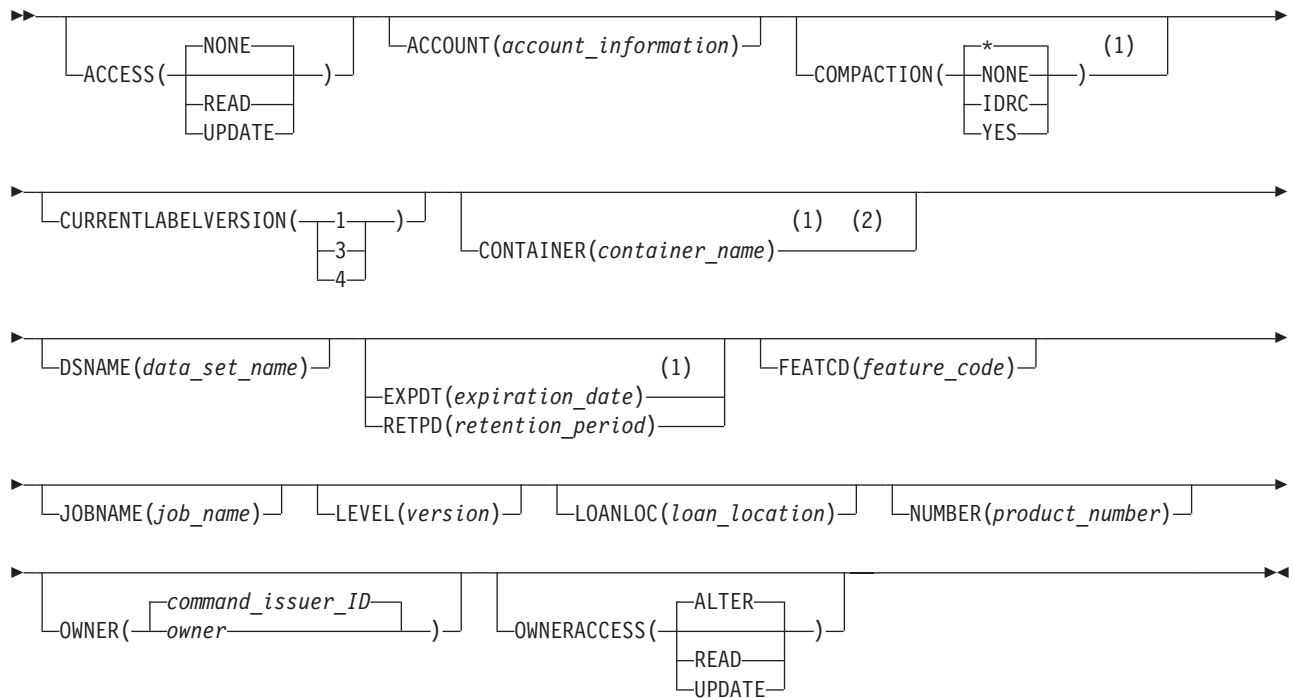


Notes:

1. Tape configuration database information used when STATUS(VOLCAT) is specified.
2. Import/Export support is available with APAR OW36342 or OW36343.

Figure 142. ADDVOLUME Syntax Diagram

Syntax



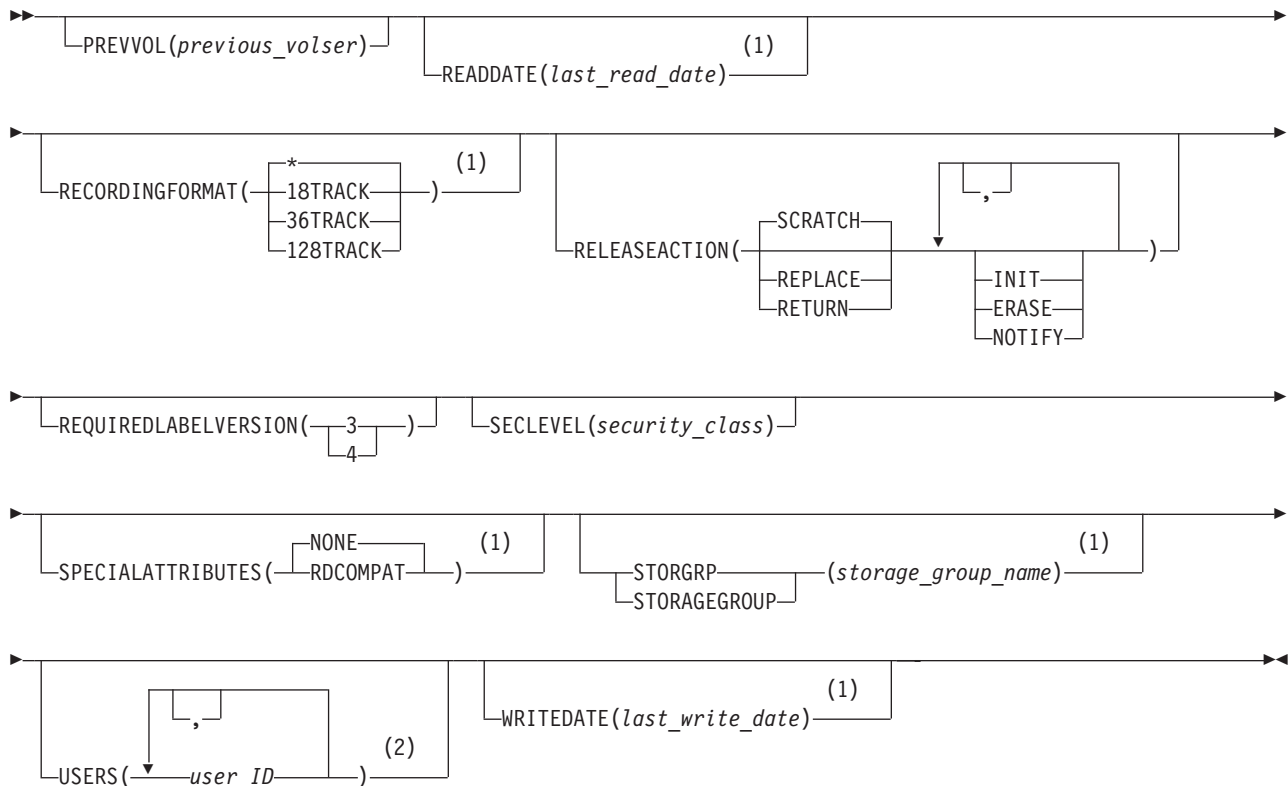
Notes:

1. Tape configuration database information used when STATUS(VOLCAT) is specified.
2. Import/Export support is available with APAR OW36342 or OW36343.

Figure 143. ADDVOLUME Non-SCRATCH Parameters

ADDVOLUME Subcommand

Syntax



Notes:

1. Tape configuration database information used when STATUS(VOLCAT) is specified.
2. You can specify a maximum of 12 user IDs.
3. Import/Export support is available with APAR OW36342 or OW36343.

Figure 144. ADDVOLUME Non-SCRATCH Parameters (Cont.)

Operands

ACCESS(NONE,READ,UPDATE)

Specifies user access to a volume. Supply a value to define the access level for users defined in the list of users who can access this volume (USERS). You can use one of the following:

NONE users do not have access to the volume

READ users have only read access to the volume

UPDATE

users have write access to the volume

The default is NONE. This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

ACCOUNT(account_information)

Specifies accounting information. Accounting information is one to forty characters enclosed in single quotes if it contains any special characters or blanks.

ADDVOLUME Subcommand

If you do not use ACCOUNT, DFSMSrmm obtains the information when it records information about the first data set on the volume. At that time, DFSMSrmm gets the accounting information from either the account number of the job or job step that creates the first data set, or from the account number of the job that reads the data set.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

ASDATE(*assigned_date*)

For a master or user volume, ASDATE specifies the date when the volume was assigned to a user. For a scratch volume, ASDATE specifies the date when the volume returned to scratch status. Supply the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 93001.
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm defaults to the 20th century.

The default is the date you issue the ADDVOLUME subcommand.

ASTIME(*assigned_time*)

Specifies the time the volume was assigned to a user

For a master or user volume, ASTIME specifies the time when the volume was assigned to a user. For a scratch volume, ASTIME specifies the time when the volume returned to scratch status. ASTIME format is *hhmmss* where:

- *hh* is hours
- *mm* is minutes
- *ss* is seconds

For example, nine o'clock in the morning is 090000.

The default is the time when you issue the ADDVOLUME subcommand.

COMPACTION(*,**NONE**,**IDRC**,**YES**)

Specifies the compaction technique used to record data on tape volumes. Use one of the following:

- ***** The compaction is not known; or the volume is not a tape volume, and compaction does not apply. This is the default.
- **NONE** No compaction was used to record data on the volume.
- **IDRC** IDRC compaction which DFSMSrmm displays as a compaction value of YES was used.
- **YES** The data on the master or user tape volumes being added is compacted.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

ADDVOLUME Subcommand

CONTAINER(*container_name*)¹⁰

Use this operand to define a volume as an exported logical volume. Specify the volser of the stacked volume as the *container_name* if you define an exported logical volume. The value can be any alphanumeric or special characters up to 16 characters in length.

When you specify a *container_name*, DFSMSrmm sets the default volume type for the volume as TYPE(LOGICAL). If the volume type is TYPE(PHYSICAL), you must change the volume type to TYPE(LOGICAL) before volume import processing can start.

CURRENTLABELVERSION(1,3,or 4)

Specifies the ISO/ANSI label version for the volume that you are adding.

There is no default.

COUNT(*number_of_volumes*)

Specifies the number of volumes to be added. The maximum allowable decimal value is 99999.

The default is 1.

DENSITY(*,1600,3480,6250)

Specifies the volume's recording density. For a 3420 tape reel, you can use DENSITY as 1600 or 6250. For a 3480 tape cartridge, use a value of 3480. Use an asterisk if you do not know the density.

DESCRIPTION(*text*)

Specifies descriptive text about the volume. Descriptive text is one to thirty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

DSNAME(*data_set_name*)

Specifies the name of the first data set on the volume. The name follows standard MVS naming conventions for data sets. The data set name must not include a member name.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

EXPDT(*expiration_date*)

Specifies the date the volume should be considered for release. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm defaults to the 20th century. The dates 99365 and 99366 can be used only when the MAXRETPD NOLIMIT value is specified in parmlib.

EXPDT is mutually exclusive with RETPD. If you do not specify EXPDT or RETPD, DFSMSrmm uses the default retention defined in the DFSMSrmm EDGRMMxx parmlib member.

10. Import/Export support is available with APAR OW36342 or OW36343.

ADDVOLUME Subcommand

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

FEATCD(*feature_code*)

Specifies the software product's feature code on the volume. A feature code is one to four alphanumeric characters. Use this operand when you associate a volume with a software product already defined to DFSMSrmm. This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

INITIALIZE(Y,N)

Specifies whether the volume must be initialized before it can be used.

Specify Y to request initialization. Specify N to indicate that the volume does not need to be initialized. If you use INITIALIZE(Y), the volume is not available for use until initialization is confirmed.

You can use the INITIALIZE operand regardless of the volume's status. DFSMSrmm accepts INIT as an abbreviation.

If you request initialization for a scratch volume, and the initialize action is still pending when you enter the volume into an automated tape library dataserver, DFSMSrmm defers initialization to DFSMSdfp labeling support. If the volume is later ejected without being initialized, DFSMSrmm reinstates the initialize action. The default is N if you use STATUS(USER), STATUS(MASTER), or STATUS(VOLCAT). If you use STATUS(SCRATCH), the default is Y.

JOBNAME(*job_name*)

Specifies the name of the job that created the first data set on the volume. A job name is one to eight alphanumeric characters or \$, #, or @. The job name must start with an alphabetic character, \$, #, or @. You cannot use a generic jobname. Any jobname you use must be specific.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

LABEL(SL,NL,AL)

Specifies the volume's label type, which can be:

SL

IBM standard labels

NL

No label

AL

ISO/ANSI labels

SL is the default.

Use the LISTVOLUME subcommand to obtain label information for a volume. DFSMSrmm automatically records label type when a data set on the volume is opened. See "LISTVOLUME: Displaying Information about a Volume" on page 351 for more information.

LEVEL(*version*)

Specifies the version of the software product on the volume. Supply the version in the form, VnnRnnMnn, indicating the version, release and modification level. 'nn' is two numbers in the range 00 to 99. Use this operand to associate a volume with a software product already defined to DFSMSrmm.

ADDVOLUME Subcommand

The default value is V01R01M00, Version 1, Release 1, Modification 0. This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

LOANLOC(*loan_location*)

Specifies the location where the volume resides other than in the removable media library or a storage location. A loan location is one to eight characters enclosed in single quotes if it contains any special characters or blanks. For example, if you remove the volume from the removable media library and are storing it in your office, you can use your owner ID as the LOANLOC value to let others know where the volume is.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

LOCATION(**SHELF**,*library_name*)

Specifies where the volume is stored and sets the home location for the volume. Use one of the following:

SHELF

Indicates that the volume is stored in a shelf location in a non-system-managed library.

library_name

Indicates the system-managed library in which the volume is stored. This library can be either a manual tape library dataserver, or an automated tape library such as the 3495 Tape Library Dataserver. A library name is one to eight alphanumeric characters, \$, #, or @, starting with a non-numeric character.

DFSMSrmm validates this library name by ensuring that the library has been defined in the TCDB. If the library is a manual tape library dataserver, DFSMSrmm adds the volume to the TCDB; if the library is an automated tape library dataserver and the volume is not currently resident in that library, DFSMSrmm sets the volume move in progress to get the volume moved to the automated tape library dataserver; it does not add the volume to the TCDB since the system does this when the volume enters the automated tape library dataserver.

If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

MEDIANAME(*media_name*)

Describes the shape or type of a volume. Media names are defined by your installation and must be one to eight characters.

The default value is 3480.

MEDIATYPE(*,**CST**,**ECCST**,**EHPCT**,**HPCT**)

Specifies the volume's physical media type. Use one of the following:

* The volume is not a cartridge. This is the default.

CST Cartridge System Tape

ECCST

Enhanced Capacity Cartridge System Tape

EHPCT

Reserved for Extended High Performance Cartridge Tape

HPCT High Performance Cartridge Tape

ADDVOLUME Subcommand

If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

NUMBER(*product_number*)

Specifies the number of the software product associated with the volume. A software product number is one to eight characters enclosed in single quotes if it contains any special characters or blanks. Use this operand when you associate a volume with a software product already defined to DFSMSrmm.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

OWNER(*owner*)

Specifies the owner ID of the volume's owner. An owner ID is one to eight alphanumeric characters, \$, #, or @; normally a RACF user ID or RACF group name. The default is the user ID of the command issuer.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

OWNERACCESS(ALTER,READ,UPDATE)

Specifies the type of access the owner has to the volume. Depending upon your installation's RACF and DFSMSrmm security options, you can use OWNERACCESS with OWNER to define the initial RACF volume profile access. The value can be ALTER, READ, or UPDATE.

The default value is ALTER.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

OWNERACCESS(ALTER,READ,UPDATE)

Specifies the type of access the owner has to the volume. Depending upon your installation's RACF and DFSMSrmm security options, you can use OWNERACCESS with OWNER to define the initial RACF volume profile access. The value can be ALTER, READ, or UPDATE.

The default value is ALTER.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

POOL(*pool_prefix*)

Specifies a pool prefix where DFSMSrmm stores the volume in the removable media library. The value is one to five alphanumeric characters followed by an asterisk.

Pool prefixes are defined by your installation. You can view information about your pools by using the LISTCONTROL subcommand with the VLPOOL operand. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information.

If you do not supply either a pool prefix or a rack number, DFSMSrmm assigns the volume a rack number matching the volume serial number. Do not use a pool prefix if you are adding the volume to an automated tape library, because the external volume serial number must match the internal volume serial number in an automated tape library dataserver. POOL is mutually exclusive with RACK. You cannot specify POOL when TYPE(LOGICAL) is specified.

PREVVOL(*previous_volser*)

Specifies the volume serial number of the previous volume for a multivolume data set. A previous volume serial number is one to six alphanumeric characters, \$, #, or @.

ADDVOLUME Subcommand

Note: You must add the volumes in a multivolume data set in their correct sequence. You must add the first volume of a multivolume data set before you can add the rest of the volumes.

RACK(*rack_number*)

Specifies a shelf location in the removable media library where DFSMSrmm stores the volume. A rack number is a full rack number consisting of six alphanumeric or national characters. The rack number you use must be previously defined and empty.

If you do not supply a pool prefix or a rack number, DFSMSrmm assigns the volume a rack number matching the volume serial number. RACK is mutually exclusive with POOL. RACK cannot be specified when TYPE(LOGICAL) is specified.

READDATE(*last_read_date*)

Specifies when the volume was last read. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm assumes the date refers to the 20th century.

There is no default.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

RECORDINGFORMAT(*,18TRACK,36TRACK,128TRACK)

Specifies the basic recording format for tape volumes.

- * An asterisk indicates that the format is unknown or that the volume is not a tape volume. This is the default.

18TRACK

Data has been written to the volume in 18TRACKs.

36TRACK

Data has been written to the volume in 36TRACKs.

128TRACK

Data has been written to the volume in 128TRACKs. Recording format 128TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB. For scratch volumes, DFSMSrmm lets the system set this value when the volume is first used.

RELEASEACTION(SCRATCH,REPLACE,RETURN,INIT,ERASE,NOTIFY)

Specifies the action to be taken when the volume is eligible for release.

RELEASEACTION can be given as a list of keywords separated by commas.

ADDVOLUME Subcommand

The first operand describes what should happen to the volume when it is released and can be one of the following:

SCRATCH

To indicate that the volume should be returned to scratch status. SCRATCH is mutually exclusive with RETURN.

REPLACE

To indicate that the volume should be replaced with a new volume and returned to scratch status.

RETURN

To indicate that the volume should be returned to its owner. RETURN is mutually exclusive with SCRATCH.

The default is SCRATCH.

After the first operand, you can specify actions to be performed for the released volume. You can use one or more of the following operands, separated from the first operand and from each other by commas.

INIT

To request that DFSMSRmm initialize the volume.

ERASE

To request that DFSMSRmm erase the volume.

NOTIFY

To request that DFSMSRmm notify the owner that the volume is being released.

For example, you can request that DFSMSRmm notify you when it is releasing your volume, and that the volume be initialized and returned to scratch by using the following operands:

```
RELEASEACTION(SCRATCH,INIT,NOTIFY)
```

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

REQUIREDLABELVERSION(3,4)

Specifies the ISO/ANSI label version to be used in the VOL1 label for the volume when creating or rewriting the volume labels on an AL type volume.

There is no default.

RETPD(*retention_period*)

Specifies the number of days DFSMSRmm retains the volume before considering it for release. The value is one to four decimal digits. The retention period is added to today's date to compute the date the volume is to be considered for release. It cannot exceed the maximum retention period (MAXRETPD) set by your installation in parmlib member EDGRMMxx.

RETPD is mutually exclusive with EXPDT. If you do not specify EXPDT or RETPD, DFSMSRmm uses the default retention defined in the DFSMSRmm EDGRMMxx parmlib member.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

SECLEVEL(*security_class*)

Specifies the volume's security class. This value is one to eight characters, and must be one defined for your installation.

ADDVOLUME Subcommand

You can use the LISTCONTROL subcommand with the SECCLS operand to display the security classes defined for your location. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333 for more information.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

SPECIALATTRIBUTES(NONE,RDCOMPAT)

Specifies any special attributes associated with the tape volume.

NONE

The tape volume has no special attributes.

RDCOMPAT

The tape volume was created using one format and can be mounted on a drive that supports reading but not writing of that format.

For example, a volume recorded at 18TRACK can be read by a device that writes at 36TRACK and also has the ability to read 18TRACK tape volumes.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

STATUS(SCRATCH,MASTER,USER,VOLCAT)

Specifies the volume's status. Use one of the following:

SCRATCH

Indicates that the volume is free and available for use.

MASTER

Indicates that this is a private volume which cannot be overwritten unless the data set names match.

USER

Indicates that this is a private volume which can be overwritten by any data set.

VOLCAT

Specifies that DFSMSrmm obtain the volume's status from the TCDB. If the TCDB contains no record for a volume, DFSMSrmm sets the volume's status to scratch. If you use any ADDVOLUME operands that do not apply to the volume's status as defined in the TCDB, DFSMSrmm overrides those operands with values in the TCDB. DFSMSrmm uses any ADDVOLUME operands that you give that apply to the volume's status to supplement information from the TCDB. If you do not use STATUS(VOLCAT), the values you provide using the ADDVOLUME subcommand are used to override values in the TCDB, except for STATUS and STORGRP values which can never be used to override values in the TCDB.

STORAGEGROUP/STORGRP(*storage_group_name*)

Specifies the SMS-defined storage group to which the volume belongs. A storage group name is one to eight alphanumeric characters. DFSMSrmm validates this name against the current SMS configuration by ensuring that the storage group has been defined.

ADDVOLUME Subcommand

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB. It is ignored if the volume is already defined in a TCDB with a different storage group name.

TYPE(LOGICAL,PHYSICAL)¹¹

Use the TYPE operand to identify a volume as either a logical or physical volume. If TYPE is not specified, DFSMSrmm sets the default value based on the LOCATION specified. If the location is a VTS, DFSMSrmm sets TYPE(LOGICAL) as the default. If the location is not a VTS, DFSMSrmm sets TYPE(PHYSICAL) as the default.

USE(MVS,VM)

Specifies the operating systems where the volume can be used. You can select MVS, VM or both. To indicate both, enter MVS and VM with a comma as a separator. The default is MVS.

USERS(*user_ID,user_ID...*)

Specifies the user IDs and group names of users that are allowed to access the volume. The type of access they have is defined by the ACCESS operand. You can supply a maximum of twelve user IDs, separated by blanks or commas.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

volser

Specifies the volume serial number. A volume serial is one to six alphanumeric characters, \$, #, or @, or special characters. You cannot use a generic volume serial number. If the volume serial number you supply is less than six characters, you must specify either a rack number or a pool prefix for the volume.

volser is required and must follow the ADDVOLUME subcommand.

WRITEDATE(*last_write_date*)

Specifies when the volume was last written to. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm assumes it refers to the 20th century.

There is no default.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

11. Import/Export support is available with APAR OW36342 or OW36343.

ADDVOLUME Subcommand

Examples

Task: Add a 3480 tape cartridge to the removable media library. The volume is a user volume, belonging to the owner whose user ID is GOHRB, and is to be used only on MVS systems. The volume serial number is 8E1U01. The volume should reside in pool U* with a media name of 3480.

Command:

```
RMM ADDVOLUME 8E1U01 DENSITY(3480) OWNER(GOHRB) -  
STATUS(USER) USE(MVS) POOL(U*) MEDIANAME(3480) MEDIATYPE(CST)
```

or, as MVS is a default value, you can enter:

```
RMM ADDVOLUME 8E1U01 DENSITY(3480) OWNER(GOHRB) -  
STATUS(USER) MEDIANAME(3480) POOL(U*) MEDIATYPE(CST)
```

Task: Define 1000 volumes in a system-managed tape library.

Command:

```
RMM ADDVOLUME A00000 COUNT(1000) LOCATION(LIB1) STATUS(SCRATCH) -  
MEDIATYPE(CST)
```

Task: Add 500 volumes, using status from the TCDB.

Command:

```
RMM ADDVOLUME SM0000 COUNT(500) STATUS(VOLCAT)
```

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDVRS: Adding a Vital Record Specification

Use the ADDVRS subcommand as shown in Figure 145 on page 257 to add a new vital record specification to DFSMSrmm. A vital record specification is used to define retention and movement policies for data sets and volumes.

You can specify any location except SHELF in a vital record specification. If you want to move a volume to SHELF, you can move the volume only if the volume's home location is SHELF. See "Chapter 6. Retaining and Moving Your Volumes" on page 99 for more information.

Your installation can use both DFSMSrmm built-in storage locations and installation defined storage locations. The built-in storage location names are LOCAL, DISTANT, and REMOTE. Installation defined storage locations are names up to eight characters long defined using the LOCDEF command in the DFSMSrmm parmlib member. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information.

DFSMSrmm applies policies to data sets and volumes when inventory management vital record processing is run. Your installation controls how DFSMSrmm applies policies based on the parmlib OPTION VRSEL operand value. See "Which Type of Vital Record Processing Should You Use?" on page 130 for information about how the VRSEL operand value effects processing. During inventory management vital record processing, DFSMSrmm matches vital record specifications with data sets and volumes to determine retention and movement. If two or more data sets on a volume match a vital record specification, there can be a conflict in the location where the volume should move. DFSMSrmm moves the volume based on the storage location priority. DFSMSrmm uses a default priority or one your installation defines using the DFSMSrmm parmlib member EDGRMMxx LOCDEF command. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information.

Data Set Vital Record Specifications

Using the ADDVRS with the DSNNAME operand and either specific data set names or data set name masks, you can define vital record specifications for data sets or groups of data sets. You can also specify the JOBNAME operand to match on the name of the job that created the data set. A retention policy is a complete vital record specification chain that includes one or more vital record specifications linked together.

You can specify one or more retention policies in a vital record specification. When only one policy applies to a data set, it is the only factor controlling the retention of the data set or volume. When you use multiple retention policies, all the conditions must be true for the data set to be retained by the vital record specification or vital record specification subchain.

You can define data set vital record specifications for tape data sets that use special expiration dates in JCL to define management and retention requirements. You use a management class name and a vital record specification management value, assigned by your installation, instead of a data set name in the data set vital record specification.

See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information on assigning management class names and vital record specification management values.

ADDVRS Subcommand

Name Vital Record Specifications

Use the ADDVRS subcommand with the NAME operand to define retention and movement policies by linking name vital record specifications to data set or volume vital record specifications. Although name vital record specifications can define both retention and movement policies, DFSMSrmm only processes the retention information when your installation has specified the DFSMSrmm EDGRMMxx parmlib OPTION VRSEL(NEW) operand. You can link as many vital record specifications together as you want.

Volume Vital Record Specifications

Use the ADDVRS subcommand with the VOLUME operand to define a vital record specification for a volume, supplying the volume's serial number.

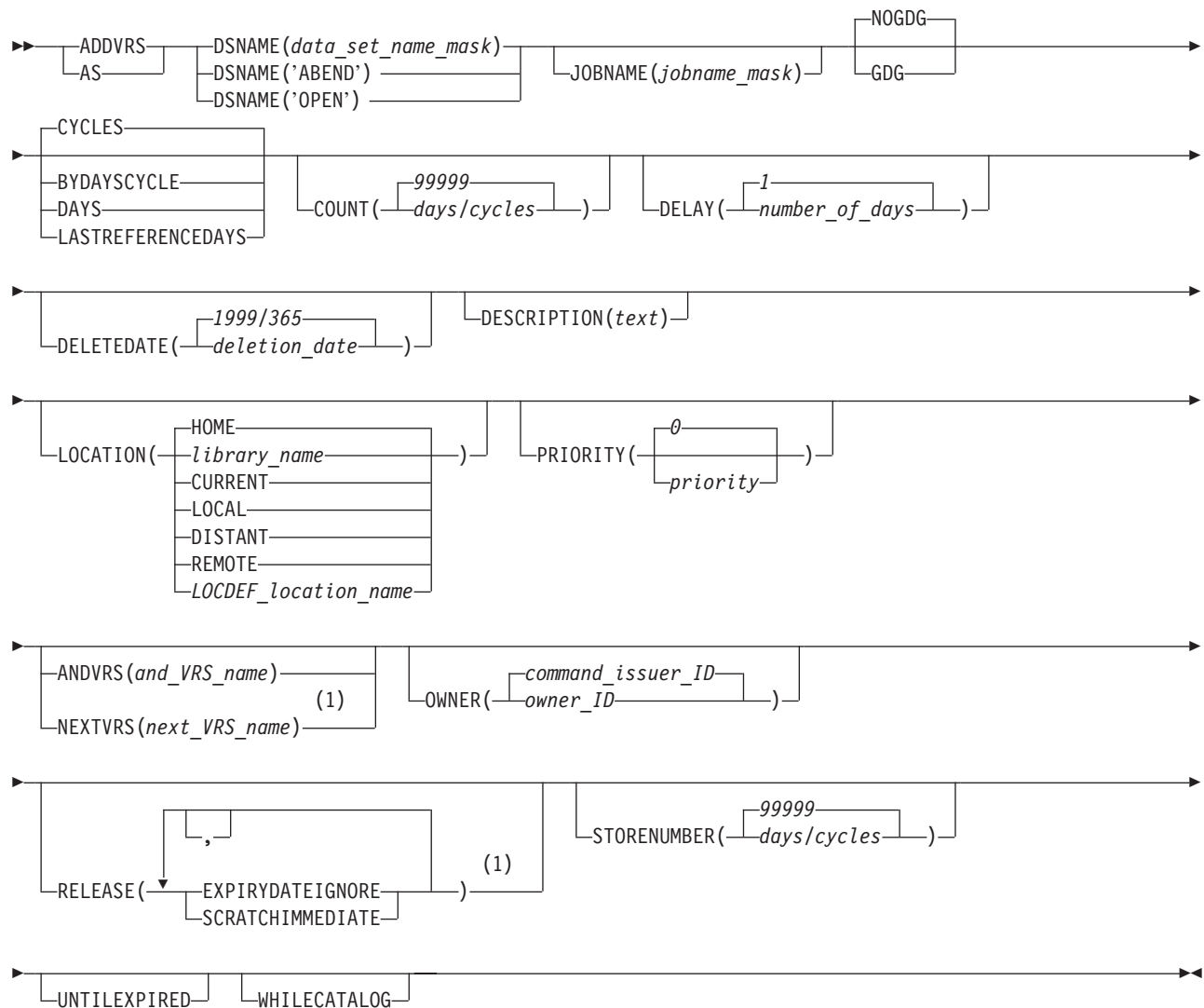
Example

Task: Add a data set vital record specification to retain all DFSMSHsm backup and migration tapes in a library called LIB1. BPREF and MPREF are the defined qualifiers for DFSMSHsm.

Command:

```
RMM ADDVRS DSN('*PREF.%%TAPES.DATASET') COUNT(99999)-  
LOCATION(LIB1)
```


Syntax

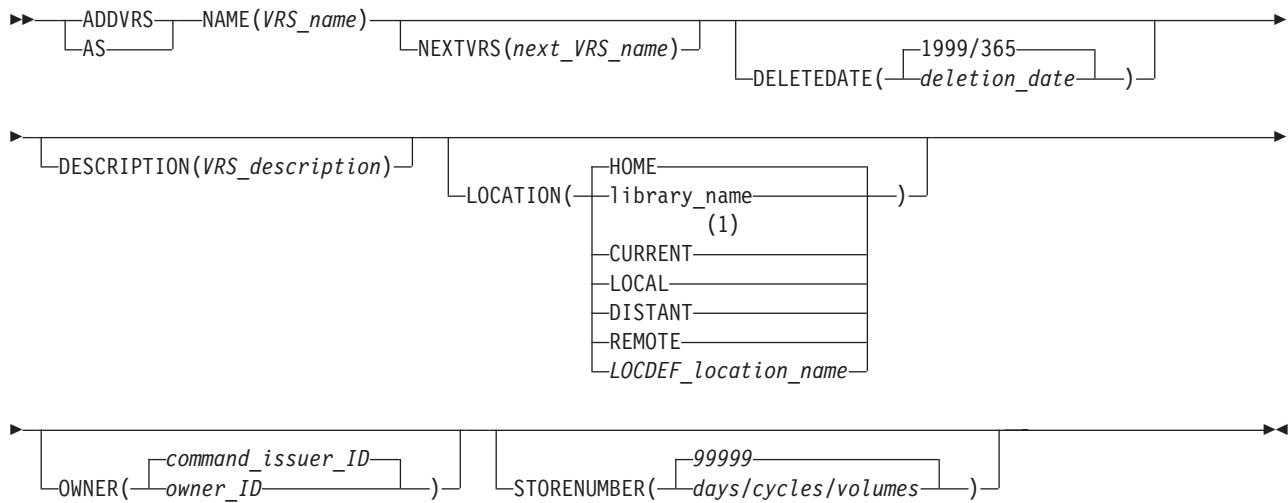


Notes:

1. Honored when DFSMSrmm parmlib OPTION VRSEL(NEW) operand is specified.

Figure 145. ADDVRS DATASET Syntax Diagram

ADDVRS Subcommand

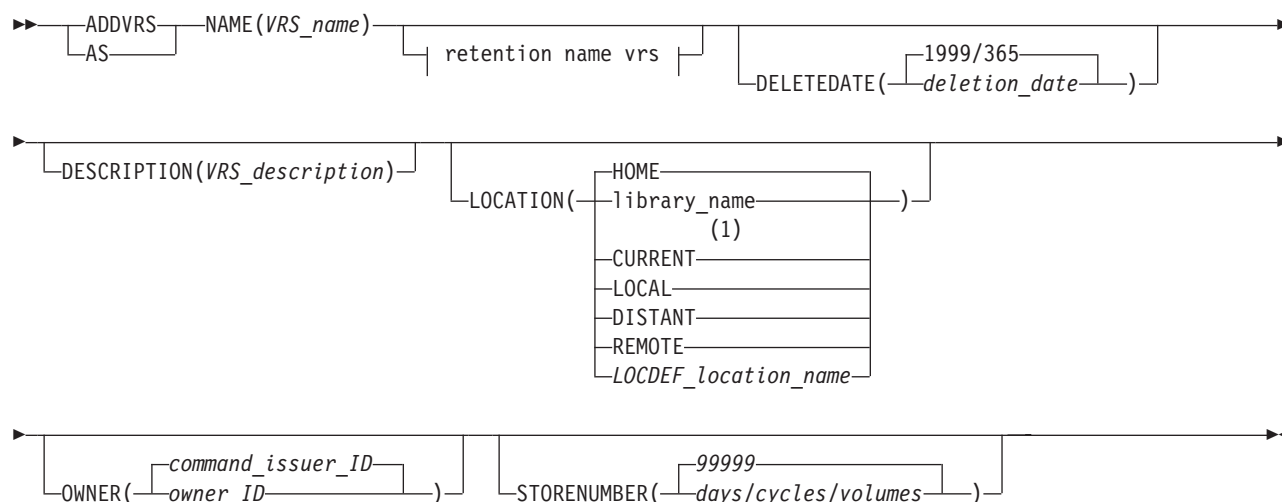


Notes:

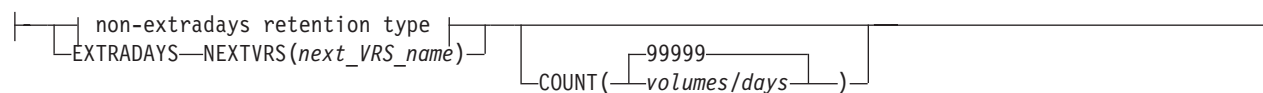
1. Honored when DFSMSrmm parmlib OPTION VRSEL(NEW) operand is specified.

Figure 146. ADDVRS Location NAME Syntax Diagram

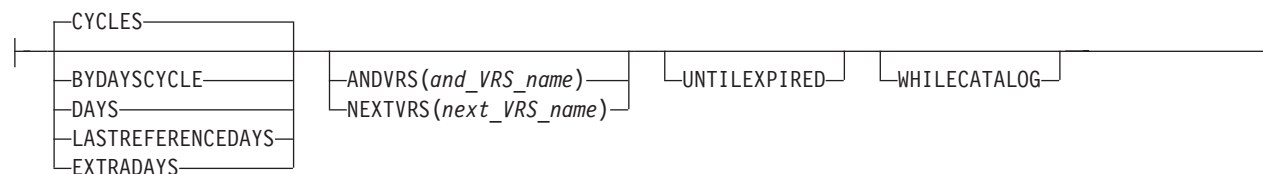
ADDVRS Subcommand



retention name vrs:



non-extradays retention type:



Notes:

1. Honored when DFSMSrmm parmlib OPTION VRSEL(NEW) operand is specified.

Figure 147. ADDVRS Retention NAME Syntax Diagram

ADDVRS Subcommand

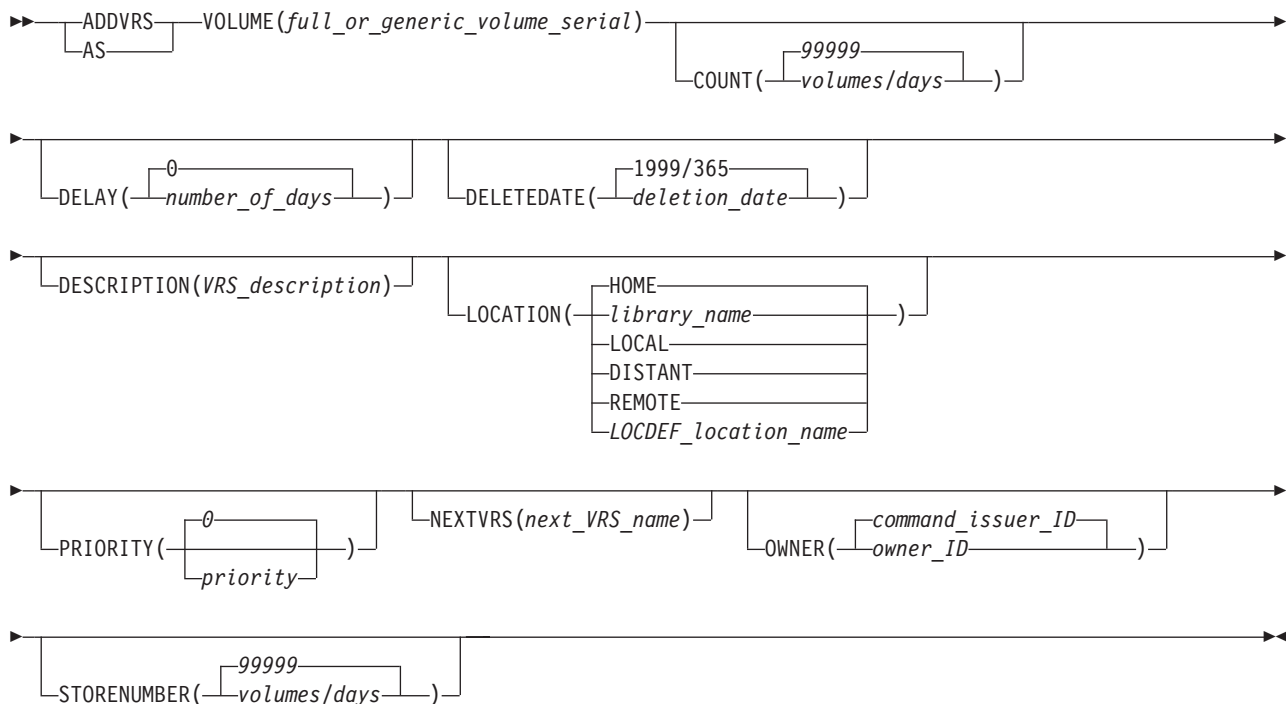


Figure 148. ADDVRS VOLUME Syntax Diagram

Operands

ANDVRS(*and_VRS_name*)

Specify ANDVRS to create a chain of vital record specifications. The chain contains all the retention conditions that must be true to retain the data set. *and_VRS_name* specifies the name of the vital record specification that links other vital record specifications in a chain. DFSMSrmm uses the STORENUMBER and LOCATION from the first vital record specification in the ANDVRS subchain. If your installation does not define parmlib OPTION VRSEL(NEW), DFSMSrmm ignores the retention information in the vital record specification.

BYDAYSCYCLE

Specify to retain all instances of a data set created on a single day as a single cycle. BYDAYSCYCLE is mutually exclusive with CYCLES, DAYS, LASTREFERENCEDAYS, and EXTRADAYS. BYDAYSCYCLE can be used on data set vital record specifications and name vital record specifications.

COUNT(*days/cycles/volumes*)

Specifies a retention amount, based on the retention type chosen: number of days or cycles for data sets, number of volumes for volumes.

Use COUNT(*number_of_days*) to request that DFSMSrmm retain all cycles or copies of a data set, or a volume for the number of days you require.

Use COUNT(*number_of_cycles*) to request that DFSMSrmm retain the number of data set cycles you want.

Use COUNT(*number_of_volumes*) to request that DFSMSrmm retain the number of volumes you want.

ADDVRS Subcommand

The value is in the range 1 to 99999. A value of 99999 indicates that DFSMSrmm retains all cycles of a data set, or all volumes specified. The default value is COUNT(99999) except when a name vital record specification is used and specifies a retention type.

When you use the DELAY operand with the DAYS and LASTREFERENCEDAYS operands, the COUNT value you specify must be greater than or equal to the total of the DELAY value and the STORENUMBER value. For the CYCLES and BYDAYSCYCLE operands, the COUNT value must be greater than or equal to the STORENUMBER value.

On a NAME VRS, the COUNT default is zero unless a retention type is specified. If a retention type is specified on the NAME VRS, the COUNT value is 999999. On all other vital record specifications, the COUNT default is 99999.

DFSMSrmm validates the COUNT value as follows:

- COUNT must be equal to STORENUMBER when the EXTRADAYS operand is specified.
- COUNT can be greater than or equal to STORENUMBER independent of specifying the NEXTVRS or ANDVRS operands.
- COUNT must be greater than or equal to STORENUMBER when DAYS or LASTREFERENCEDAYS retention is used.

CYCLES

Specifies that DFSMSrmm retains data sets based on cycles or copies of a data set. For GDG data set vital record specifications this is based on GDG cycles. For non-GDG data sets, each occurrence of a data set is considered to be a cycle. CYCLES can be used for data set vital record specifications and name vital record specifications. CYCLES is mutually exclusive with DAYS, LASTREFERENCEDAYS, BYDAYSCYCLE, and EXTRADAYS.

CYCLES is the default for data set vital record specifications.

CYCLES is the default for name set vital record specifications if COUNT or ANDVRS is specified but no retention type is specified.

DAYS

Specifies that DFSMSrmm retains all data sets based on a number of days since creation. For example, ADDVRS ... DAYS COUNT(5), requests that DFSMSrmm retain all data sets created during the last five days. DAYS can be used for both data set vital record specifications and name vital record specifications. DAYS is mutually exclusive with CYCLES, LASTREFERENCEDAYS, BYDAYSCYCLE, and EXTRADAYS.

DELAY(*number_of_days*)

Specifies the number of elapsed calendar days (even if retaining by cycles) that you would like DFSMSrmm to retain a volume in its current location before sending it to the location specified on the ADDVRS subcommand. DFSMSrmm calculates the date to send the volume to the specified location by adding the DELAY(*number_of_days*) to the creation date. For example, you might create a data set that matches to a vital record specification with DELAY(1), at 11:59P.M. on November 12th after running vital record processing on the 12th. If you run vital record processing at 12:01A.M. on November 13th, DFSMSrmm will move the volume to the location named on the ADDVRS subcommand.

For a data set vital record specification, DFSMSrmm uses the value you specify for DELAY to retain only the latest cycle or incidence of the data set. If you use

ADDVRS Subcommand

the DAYS operand to supply a number of days as the retention type for all cycles of the data set, and if a new cycle of the data set is created before the delay period elapses, the cycle being delayed is transferred to the first location. DFSMSrmm retains it in the location for the total of the remainder of the delay period and for the number of days you supplied for the location.

For a volume vital record specification, DFSMSrmm retains the volume regardless of any cycles of data sets that might reside on the volume. The maximum allowable decimal value is 99. The default value is 0.

If you use DELAY and retention is not by cycles, the COUNT value you use must be greater than or equal to the total of the STORENUMBER value and the DELAY value you specify. If you also supply a NEXTVRS value, the COUNT value you use must be greater than the total of the STORENUMBER value and the DELAY value you specify.

You cannot use DELAY if you specify LOCATION(HOME).

DELETEDATE(*deletion_date*)

Specifies the date when DFSMSrmm deletes the vital record specification. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm defaults to the 20th century.

Once DFSMSrmm deletes a vital record specification, all data sets or volumes retained by the vital record specification become eligible for normal expiration processing.

The default value is 1999/365, which indicates that DFSMSrmm never deletes the vital record specification.

DESCRIPTION(*text*)

Specifies descriptive text about the vital record specification. Descriptive text is one to thirty characters enclosed in single quotes if it contains any special characters or blanks. The default is blanks.

DSNAME(*data_set_name_mask*)

Identifies the type of vital record specification and gives a data set name for the vital record specification. You can supply a fully qualified data set name, a data set name mask, or a GDG data set name. Fully qualified names take precedence over data set name masks. If a ~ is used in a data set name, the name is treated as a pseudo GDG entry.

You can use an SMS management class name or a vital record specification management value. The name can be eight alphanumeric characters, beginning with an alphabetic character, and must follow standard MVS data set naming conventions. This name must be a single qualifier, and is already assigned by your installation. For example, if the management class name M99000 has been assigned

ADDVRS Subcommand

by your installation to data sets with the special date 99000, you can define a vital record specification to cover those data sets by specifying `DSNAME('M99000')` as part of the ADDVRS subcommand.

You can also use a data set name mask to define a vital record specification that matches to several management class names or vital record specification management values. For example, you could use the data set name mask `M9*` to define a vital record specification that covers any special dates in the range 98001 through 99366 that have been assigned a management class name or vital record specification management value. This data set name mask must be a single qualifier.

If a data set name mask matches a management class name or a vital record specification management value, do not specify the GDG operand.

Specify the data set name mask `ABEND` to manage all data sets closed as a result of an abnormal end in a task. You can also use the `JOBNAME` operand to manage these data sets by job name.

Specify the data set name mask `OPEN` to manage all data sets open at the time inventory management vital record processing is run. These data sets might have been left open by a system failure or might be in use. You can also use the `JOBNAME` operand to manage these data sets by job name.

You can use `*`, `%`, or `~` in a data set name mask.

- * A single `*` represents a single qualifier of any number of characters.

A single `*` when used within a qualifier represents zero or more characters.

More than one single `*` can be used within a qualifier as long as a character precedes or follows the `*`.

`.**` represents zero or more qualifiers. At the end of the mask, `**` indicates to ignore any remaining characters.

`**` indicates to select all data sets. You can use this mask to define a vital record specification that sets your installation default retention criteria for data sets that are not covered by other vital record specifications.

The data set name masks `.**` and `**` match to all data sets not covered by a more specific vital record specification. You can use these data set name masks to define a system-wide release option.

% (percent sign)

A place holder for a single character.

~ (not sign)

A place holder for a single character in a pseudo-gdg data set name mask. The `~` has special meaning in a data set name mask and is used to specify a pseudo-GDG data set name. See “Pseudo-GDG Data Set Names” on page 88 for information.

Use `%` when you do not want to manage all the data sets that match the data set name mask as a pseudo-GDG.

ADDVRS Subcommand

When defining policies to manage a GDG base entry and a standard data set name, you cannot use the same data set name in two vital record specifications. You also cannot have two vital record specifications that use the same data set name and job name combination when managing GDGs and non-GDGs. If you want to define different retention and movement criteria for a data set name, use the GDG suffix together with `~` in the data set name for one of the vital record specifications. For example, you can define the following vital record specifications for the same data set name:

```
RMM ADDVRS DSNAME(a.b.c) NOGDG CYCLE COUNT(5) LOCATION(REMOTE)
RMM ADDVRS DSNAME(a.b.c.G---V---) -
    NOGDG CYCLES COUNT(5)
```

DFSMSrmm manages both sets of non-GDG data sets separately; because you include the GDG suffix and `~` in the data set name for the second vital record specification, DFSMSrmm manages those data sets as a cycle of pseudo-GDG data sets, even though they are identified as NOGDG.

You must select a vital record specification type of DSNAME, VOLUME, or NAME.

EXTRADAYS

Specifies the number of days since a name vital record specification started to retain the data set. The number of extra days is specified with the COUNT operand. EXTRADAYS is the number of days since the NAME vital record specification started to retain the data set. The number of days depends on when the previous vital record specification stopped retaining the data set and the time when vital record processing is run. EXTRADAYS can only be used in a NAME vital record specification. EXTRADAYS cannot be used with CYCLES, DAYS, LASTREFERENCEDAYS, BYDAYSCYCLE, WHILECATALOG, UNTILEXPIRED, or ANDVRS.

GDG

Specifies that the data set name is a GDG base name. For GDG data set vital record specifications DFSMSrmm checks for the standard version qualifier GnnnnVnn. GDG is only valid for a data set name vital record specification. If you do not use GDG, the default NOGDG is in effect.

JOBNAME(jobname_mask)

Specifies a job name that created a data set. A job name is one to eight alphanumeric characters or \$, #, or @. The job name must start with an alphabetic character, \$, #, or @. You can specify a specific jobname or a jobname mask. Use % to match any one character and * to match any character string in the mask. Specific job names take precedence over job name masks. This operand is optional. DFSMSrmm applies JOBNAME in vital record processing based on the parmlib OPTION command VRSJOBNAME operand as described in the *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*. If JOBNAME is not specified, vital record processing is based on the data set name only.

LASTREFERENCEDAYS

Specifies that DFSMSrmm retains all copies of the data set based on the number of elapsed days since the data set was last read or written to.

LASTREFERENCEDAYS can be used for data set vital record specifications and name vital record specifications. LASTREFERENCEDAYS is mutually exclusive with CYCLES, DAYS, BYDAYSCYCLE, and EXTRADAYS.

LOCATION(*location_name*)

Specifies a location where the volume should be retained. *location_name* can be: CURRENT,HOME,*library_name*,LOCAL,*LOCDEF_location_name*,DISTANT, or REMOTE.

Use HOME if you want the volume to be returned to its home location. If the volume's home location is defined to DFSMSrmm as SHELF, you can use HOME to return a volume to a location in a non-system-managed tape library. See "ADDVOLUME: Adding Volume Information" on page 241 and "CHANGEVOLUME: Changing Volume Information" on page 282 for information on specifying a volume's home location. Use CURRENT to avoid moving volumes.

Supply a library name if you want the volume to be retained in a system-managed tape library. A library name is one to eight alphanumeric characters, starting with a non-numeric character.

Specify an installation defined storage location name or LOCAL, DISTANT, or REMOTE if you want DFSMSrmm to retain the volume in a storage location.

The default value is HOME.

NAME(*VRS_name*)

Identifies the vital record specification type and gives a name for the vital record specification. A vital record specification name is one to eight alphanumeric or national characters chosen by your installation. DFSMSrmm folds the value to upper-case. A NAME vital record specification identifies a location and can optionally specify retention type and count.

To link vital record specifications together, use a vital record specification name as the NEXTVRS or ANDVRS value on data set, volume, and other name vital record specifications.

You must specify a vital record specification type of NAME, VOLUME or DSNAME.

NEXTVRS(*next_VRS_name*)

Specifies the name of the next vital record specification in a chain of vital record specifications.

If you give the name of a vital record specification that does not exist, DFSMSrmm returns the volume or data set to its home location when the retention criteria you specify has been met and issues message EDG2230I and return code 4.

NOGDG

Specifies that the data set name does not identify a GDG base name. For NOGDG data set vital record specifications, you can use a fully qualified generation data set name. NOGDG is only valid for data set name vital record specifications.

NOGDG is the default.

OWNER(*owner*)

Specifies the owner ID of the vital record specification's owner. An owner ID is one to eight alphanumeric characters, \$, #, or @. The owner must be previously defined to DFSMSrmm. The default is the user ID of the command issuer.

ADDVRS Subcommand

PRIORITY(0-9999)

Specifies a priority value 0 - 9999 to override the priority defined in the LOCDEF command for a location. The priority values are purely relative and do not have any further significance. The lower priority numbers take precedence. For example, in the case of a tie, DFSMSrmm requests that a volume move to the location with priority value 100 rather than priority value 200. The default value 0 means that the LOCDEF defined priority should be used.

When PRIORITY is used on a data set name vital record specification or volume vital record specification, the priority applies to the data set or volume no matter which vital record specification in a chain currently retains the data set or volume. The priority specified in the vital record specification is used instead of the priority defined by the LOCDEF parmlib command value. PRIORITY cannot be used on a NAME vital record specification.

RELEASE(EXPIRYDATEIGNORE,SCRATCHIMMEDIATE)

Specifies special release options to allow the release of data sets if the only release action for the volume is return to scratch. RELEASE options take affect only for data sets retained by a vital record specification.

EXPIRYDATEIGNORE is specified so DFSMSrmm ignores the volume expiration date when VRSEL(NEW) is used and no other vital record specification matches to the data set.

SCRATCHIMMEDIATE is specified so DFSMSrmm returns the volume to scratch status when the only pending release action for the volume is to return to scratch.

STORENUMBER(*days/cycles/volumes*)

Specifies how many days to retain a data set, how many data set cycles or versions to retain, how many volumes to retain, or how many days to retain a volume.

DFSMSrmm uses STORENUMBER(*days*) to retain a data set or a particular volume for a number of days when you add a data set vital record specification with the DAYS or LASTREFERENCEDAYS operand, or when you add a volume vital record specification for a volume.

DFSMSrmm uses STORENUMBER(*cycles*) to retain a number of data set cycles when you add a data set vital record specification with the CYCLES operand.

DFSMSrmm uses STORENUMBER(*volumes*) to retain a number of volumes when you add a vital record specification using a generic volume serial number.

The value range is 1-99999, where 99999 indicates that all remaining data sets or volumes are to be retained. The default is STORENUMBER(99999).

STORENUMBER must be less than or equal to COUNT.

UNTILEXPIRED

Specifies that DFSMSrmm retains the data set until the volume expiration date is reached or until the data set is no longer retained. When you specify UNTILEXPIRED, DFSMSrmm releases a data set as soon as the volume expiration date is reached, regardless of the overall retention amount specified by the COUNT operand, and regardless of catalog status. When

ADDVRS Subcommand

UNTILEXPIRED is specified alone, the expiration date is the only factor controlling vital records specification retention.

If a data set matches a primary vital record specification and secondary vital record specification, DFSMSrmm processing depends on the EDGRMMxx parmlib OPTION command VRSEL operand.

- If VRSEL(OLD) is in use, DFSMSrmm checks to see if the management value vital record specification includes the WHILECATALOG retention type. Specifying WHILECATALOG ensures that DFSMSrmm retains the data set as long as the data set is cataloged. If the data set is not cataloged, then DFSMSrmm retains the data set until the volume expiration date is reached.
- If VRSEL(NEW) is in use, DFSMSrmm uses the retention information from the secondary vital record specification to determine if the data set should be retained. If the secondary vital record specification retains the data set, the expiration date is used to retain the data set. If the secondary vital record specification does not retain the data set, then the primary vital record specification retains the data set.

The UNTILEXPIRED retention uses the volume expiration date when used in a management value or management class vital record specification.

See “Merging Retention Policies” on page 105 for information about DFSMSrmm processing when UNTILEXPIRED and other retention types like WHILECATALOG are used at the same time.

VOLUME(*full_or_generic_volume_serial*)

Specifies the volume serial number for the vital record specification being added. You can supply a full volume serial number or a generic volume serial number. A full volume serial number is one to six alphanumeric characters, \$, #, or @, or special characters. A generic volume serial number is one to five alphanumeric characters, \$, #, or @, or special characters followed by an asterisk.

You must use a vital record specification type of VOLUME, DSNAME or NAME.

WHILECATALOG

Specifies that DFSMSrmm retains the data set only as long as it is cataloged. When you use WHILECATALOG, DFSMSrmm does not retain a data set if it is not cataloged at the time of vital record processing. To retain all versions of a data set while they are cataloged, use ADDVRS DSN('**') WHILECATALOG COUNT(99999). If a data set is not cataloged, DFSMSrmm uses the parmlib OPTION CATRETPD operand to define a minimum catalog retention period. This prevents a volume from being released before the data set that resides on it gets cataloged at the completion of a long running job.

The default is DFSMSrmm retains the data set regardless of whether or not it is cataloged.

See “Merging Retention Policies” on page 105 for information about DFSMSrmm processing when UNTILEXPIRED and other retention types like WHILECATALOG are used at the same time.

Examples

Task: Add a volume vital record specification. The volume's owner has the user ID, OWNER77, and the volume serial number is 8E1U02. Indicate that the volume is to remain in its home location indefinitely.

ADDVRS Subcommand

Command:

```
RMM ADDVRS VOLUME(8E1U02) OWNER(OWNER77) LOCATION(HOME) COUNT(99999)
```

Task: Define two vital record specifications to use DFSMSHsm ABARS to take application backups for retention first in the LOCAL storage location, and then in the REMOTE storage location. Create a pseudo-GDG to retain the volumes by using ~ in the data set name mask. This is because DFSMSHsm uses its own version identifier in the data sets it creates.

1. The first vital record specification identifies the data sets to be retained, '*DFHSM.%.C%V~', the number of versions (5), and that the latest generation should be kept in the removable media library for one day, then moved to the LOCAL storage location.
2. The second vital record specification identifies that two versions are to be kept in the REMOTE storage location.

Command 1:

```
RMM ADDVRS DSNAME('*DFHSM.%.C%V~') COUNT(5) -  
    DELAY(1) CYCLES STORENUMBER(1) LOCATION(LOCAL) NEXTVRS(REMOTE)
```

Command 2:

```
RMM ADDVRS NAME(REMOTE) STORENUMBER(2) LOCATION(REMOTE)
```

These commands keep all your ABARS-produced backups, no matter how many copies you ask DFSMSHsm to produce, using the same vital record specification. You can replace the *DFHSM with the application aggregate names you are using.

Task: Add a vital record specification for the management value defined by your installation as D99000, used to manage data sets with the special date 99000, and not covered by a management class. You want the data sets to be retained under catalog control.

Command

```
RMM ADDVRS DSNAME('D99000') WHILECATALOG
```

Task: Retain data sets matching the DSNAME mask in location DISTANT until the volume expiration has been reached.

Command

```
RMM ADDVRS DSNAME('HLQ.**') DAYS LOCATION(DISTANT) -  
    STORENUMBER(99999) UNTILEXPIRED
```

Task: Retain data sets matching the DSNAME mask and JOBNAME mask in location DISTANT until the volume expiration has been reached.

Command

```
RMM ADDVRS DSNAME('HLQ.**') JOBNAME(S*JOB) DAYS LOCATION(DISTANT) -  
    STORENUMBER(99999) UNTILEXPIRED
```

Task: Retain data sets closed as a result ofabend processing by specifying multiple ABEND or OPEN data set vital record specifications. For example, you can define the following vital record specifications for data sets closed byabend processing:

Command

ADDVRS Subcommand

```
RMM ADDVRS DSNAME('ABEND') JOBNAME(JOB123) DAYS COUNT(10)
RMM ADDVRS DSNAME('ABEND') JOBNAME(JOB1*) DAYS COUNT(5)
RMM ADDVRS DSNAME('ABEND') DAYS COUNT(2)
```

If job JOB123 ends abnormally, all data sets closed by abend processing are retained for 10 days. If job JOB111 ends abnormally, all data sets closed by abend processing are retained for 5 days. If job JOB5678 ends abnormally, all data sets closed by abend processing are retained for 2 days.

Task: Add a volume vital record specification that will move the volume 8E1U02 directly to the storage location (STORWBIN) from its home location at the first run of inventory management vital record processing and storage location management.

Command:

```
RMM ADDVRS VOLUME(8E1U02) LOCATION(STORWBIN)
```

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

CHANGEDATASET Subcommand

CHANGEDATASET: Changing Data Set Information

Use the CHANGEDATASET subcommand as shown in Figure 149 on page 271 to update information about data sets defined to DFSMSrmm.

Give a data set name, the serial number of the volume where it resides and, optionally, the data set's relative position on the volume. DFSMSrmm assumes the data set is the first one on the specified volume unless you use the SEQ operand. DFSMSrmm issues a message if the data set with the supplied sequence number on the indicated volume is not defined to DFSMSrmm.

There are some restrictions on the information that you can change. If DFSMSrmm automatically recorded the information about the data set, you can change this information if you are authorized to use the FORCE operand. To use the FORCE operand to change data set information, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource. If you are not authorized to use the FORCE operand, the only automatically recorded information you can change is the volume's security level, job name, management class, and management value.

You cannot change the serial number of the volume where the data set resides and the data set's sequence number because they uniquely identify the data set. Use the LISTDATASET subcommand to view details about a data set, including its sequence number. See "LISTDATASET: Displaying Information about a Data Set" on page 342 for more information.

Example

Task: Update the information for a data set named PREFIX.MYDATA.DATA that resides on file 1 of volume 8E1U01. The data set on the volume has not been opened or closed so there are no restrictions on the information you can change. Change the block size to 6160 and the number of blocks to 100.

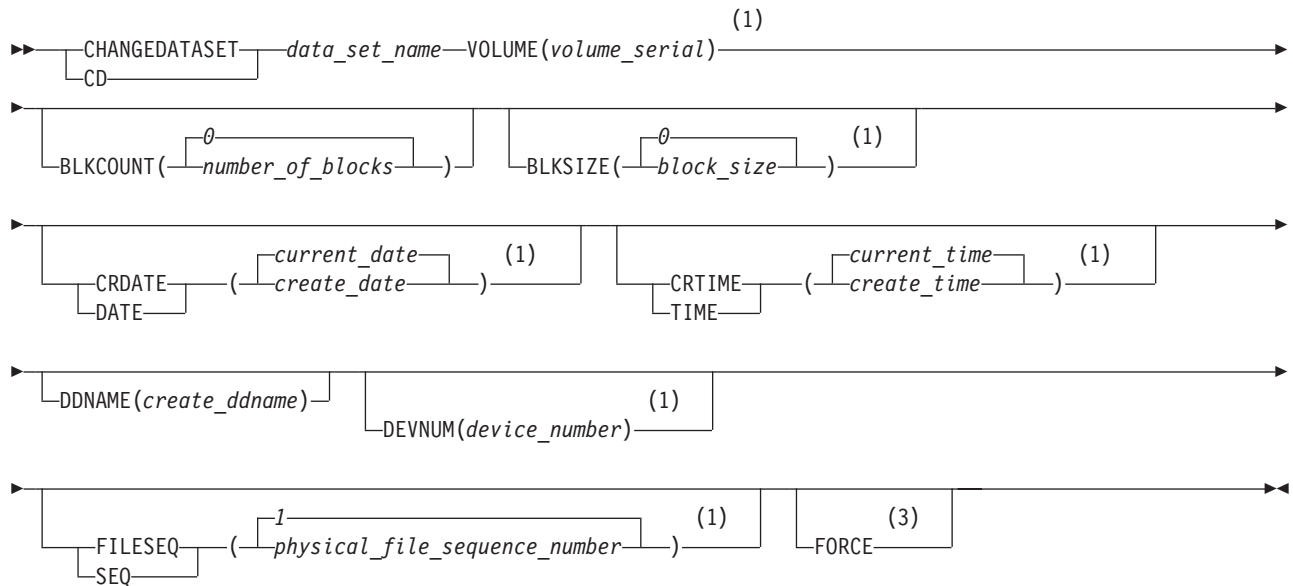
Command:

```
RMM CHANGEDATASET 'PREFIX.MYDATA.DATA' -  
VOLUME(8E1U01) BLKSIZE(6160) BLKCOUNT(100)
```

If prefix is your own TSO PROFILE PREFIX, you can enter:

```
RMM CHANGEDATASET MYDATA.DATA VOLUME(8E1U01) -  
BLKSIZE(6160) BLKCOUNT(100)
```

Syntax

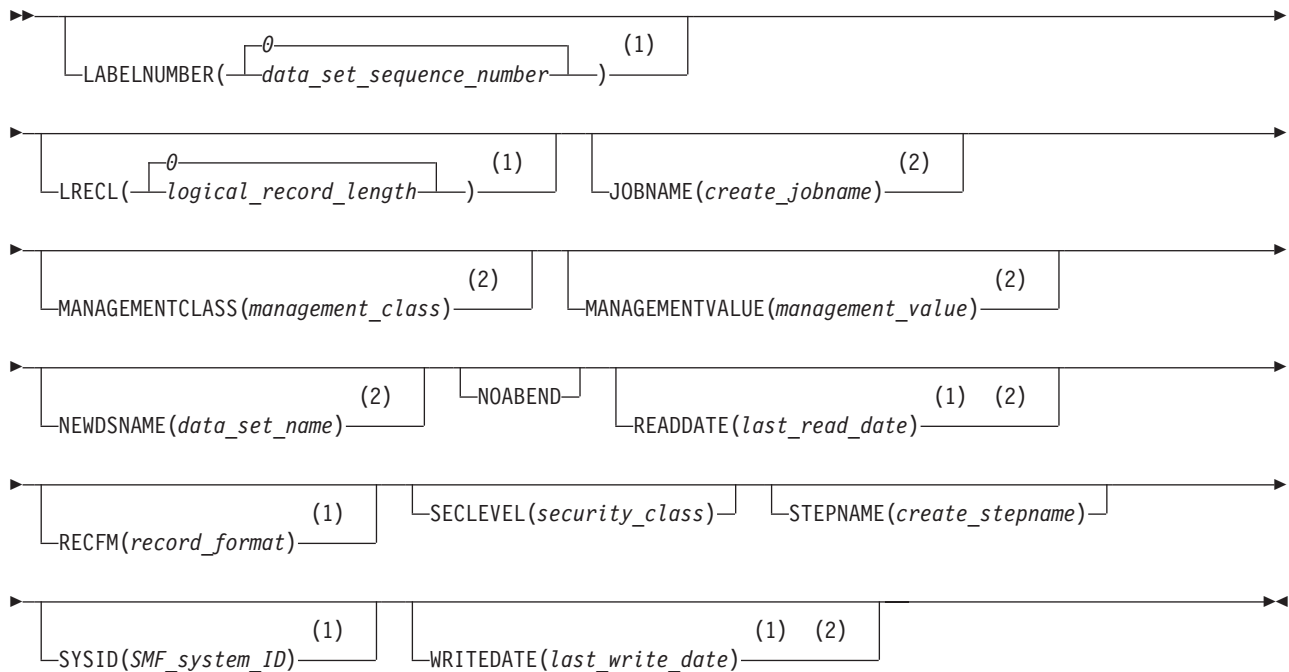


Notes:

1. Cannot be specified if DFSMSrmm automatically recorded information about the volume when a data set on the volume was first opened.
2. Can only be specified by a user with CONTROL access to STGADMIN.EDG.MASTER security resource. Owner authorization does not apply.
3. Specify to change a data set for a volume where information was recorded by DFSMSrmm during O/C/EOV processing. You must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource to use the FORCE operand.

Figure 149. CHANGEDATASET Syntax Diagram Part 1

CHANGEDATASET Subcommand



Notes:

1. Cannot be specified if DFSMSrmm automatically recorded information about the volume when a data set on the volume was first opened.
2. Can only be specified by a user with CONTROL access to STGADMIN.EDG.MASTER security resource. Owner authorization does not apply.
3. Specify to change a data set for a volume where information was recorded by DFSMSrmm during O/C/EOV processing. You must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource to use the FORCE operand.

Figure 150. CHANGEDATASET Syntax Diagram Part 2

Operands

BLKCOUNT(*number_of_blocks*)

Specify to change the number of data blocks used by the data set. The value corresponds to that recorded in the data set's End of File marker. The minimum allowable decimal value is 0; the maximum allowable decimal value is 9999999. DFSMSrmm uses BLKCOUNT together with BLKSIZE to calculate the data set's approximate size when the data set is closed. The default value is 0.

BLKSIZE(*block_size*)

Specify to change the data set's block size. The minimum allowable decimal value is 0; the maximum allowable decimal value is 32760. DFSMSrmm uses BLKSIZE together with BLKCOUNT to calculate the data set's approximate size when the data set is closed. The default value is 0.

CRDATE/DATE(*create_date*)

Specify to change the date when the data set was written to tape. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.

CHANGEDATASET Subcommand

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm defaults to the 20th century.

CRTIME/TIME(*create_time*)

Specify to change the time the data set was written to tape, in the format *hhmmss* where:

- *hh* is hours
- *mm* is minutes
- *ss* is seconds

For example, nine o'clock in the morning is 090000.

data_set_name

Specifies the name of the data set being changed. The name follows standard MVS naming conventions for data sets. The data set name must not include a member name. This operand is required and must immediately follow the CHANGEDATASET subcommand.

DDNAME(*create_ddname*)

Specifies the data description of the created data set. A DD name is one to eight alphanumeric characters, \$, #, or @. The DD name must start with an alphabetic character, \$, #, or @. You cannot use a generic DD name.

DEVNUM(*device_number*)

Specify to change the device number of the drive on which the volume was mounted when DFSMSrmm recorded information about the data set. Give a three or four character hexadecimal number, using leading zeros if the number is less than three characters.

FILESEQ/SEQ(1,*physical_file_sequence_number*)

Specifies the relative position of the data set on the volume. The minimum allowable decimal value is 1. The maximum allowable decimal value is 9999.

The default value is 1.

FORCE

Specify to override the restriction that information that DFSMSrmm recorded during O/C/EOV processing cannot be changed. Using FORCE allows you to change a data set on a volume where DFSMSrmm recorded information during O/C/EOV processing. To use the FORCE operand you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.

JOBNAME(*create_jobname*)

Specifies the job name that created the data set. A job name is one to eight alphanumeric characters or \$, #, or @. The job name must start with an alphabetic character, \$, #, or @. You cannot use a generic jobname. Any jobname you use must be specific.

When a data set has no creating job name, DFSMSrmm uses the job name that created the first file on the same volume. Use the CHANGEDATASET subcommand to set a creating job name when the job name is different than the job name used to create the first file on the volume.

CHANGEDATASET Subcommand

LABELNUMBER(*data_set_sequence_number*)

Specifies the data set sequence number you have to enter on the LABEL JCL parameter for allocating the specific data set without using the catalog entry. The minimum allowable decimal value is 0. The maximum allowable decimal value is 9999. The default value is 0. The value you specify is not validated with the values known for the preceding or following data sets on the volume.

LRECL(*logical_record_length*)

Specifies the length, in bytes, of the largest logical record in the data set. The minimum allowable decimal value is 0; the maximum allowable decimal value is 32760.

MANAGEMENTCLASS(*management_class*)

Specifies the management class name. A management class name is one to eight alphanumeric characters and must begin with an alphabetic character. The name you specify must be a single qualifier. This is an optional operand which has no default.

MANAGEMENTVALUE(*VRS_management_value*)

Specifies the vital record specification management value. A vital record specification management value is one to eight alphanumeric characters and must begin with an alphabetic character. The name you specify must be a single qualifier. This is an optional operand which has no default.

NEWDSNAME(*data_set_name*)

Specify this operand to change the name of a data set in the DFSMSrmm control data set. Note: Using this operand does not change the data set name on the physical media. DFSMSrmm associates the changed data set name with the information that is recorded for the old data set name. To change a data set name on a standard label tape, the last 17 characters of the new data set name must match the last 17 characters of the data set name recorded in the tape label. To change a data set name that was recorded by DFSMSrmm during O/C/EOV processing, you must use the FORCE operand to force the change.

This is an optional operand and has no default value.

NOABEND

Specify to reset the ABEND flag.

READDATE(*last_read_date*)

Specifies when the data set was last read. Supply the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 93001.
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm assumes the date refers to the 20th century. There is no default. If a vital record specification indicates that DFSMSrmm retain a data set by last reference days and you did not use a read or write date for the data set, DFSMSrmm uses the data set's creation date.

DFSMSrmm updates the details for the volume on which the data set resides if the last read date you use is more recent than the last read date recorded for the volume.

RECFM(*record_format*)

Specifies the format and characteristics of the records in the data sets on the volume.

U Records of undefined length

F Fixed-length record

FB

Blocked fixed-length record

FS

Fixed-length records, standard records

FBS

Fixed-length records, written as standard blocks

V Variable-length record

VB

Blocked variable-length record

VS

Variable-length records, spanned records

VBS

Variable-length records, possibly spanning more than one block

D Variable-length ASCII record

DB

Blocked variable-length ASCII record

DS

Variable-length ASCII spanned record

DBS

Variable-length ASCII blocked spanned record

You can also append either A or M to the fixed and variable formats.

A The record contains ASCII printer control characters

M The record contains machine code control characters

For example, you can use FBA or FBM.

You can append A to one of D, DB, DS, or DBS. For example, you can use DBA.

SECLEVEL(*security_class*)

Specify to change the data set's security class. The value must be one to eight characters, and must be one of the security classes defined by your installation. Use the LISTCONTROL subcommand to display your installation's security classes. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information.

STEPNAME(*create_stepname*)

Specifies the job step name of the job that created the data set. A step name is one to eight alphanumeric characters, \$, #, or @. The step name must start with an alphabetic character, \$, #, or @. You cannot use a generic step name.

SYSID(*SMF_system_ID*)

Specify to change the ID for the system where the data set was created.

CHANGEDATASET Subcommand

This can be the system ID you use for DFSMSrmm specified in EDGRMMxx parmlib member, or it can be the SMF ID for your system if you have not supplied a DFSMSrmm system identifier. The value must be one to eight alphanumeric characters, or \$, #, or @, or special characters.

VOLUME(*volume_serial*)

Specifies the serial number of the volume where the data set resides. A volume serial is one to six alphanumeric characters, or \$, #, or @, or special characters. This operand is required.

WRITEDATE(*last_write_date*)

Specifies when the data set was last written to. Supply the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 93001.
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm assumes the date refers to the 20th century.

There is no default. If a vital record specification indicates that DFSMSrmm retain a data set by last reference days and you did not use a read or write date for the data set, DFSMSrmm uses the data set's creation date.

DFSMSrmm updates the details for the volume on which the data set resides if the last write date you use is more recent than the last write date recorded for the volume.

Example

Task: Change the MANAGEMENTVALUE assigned to a data set and set the creating job name.

Command:

```
RMM CHANGEDATASET 'KEEP.UKRM.TEAM' VOLUME(OHN095) -  
MANAGEMENTVALUE(D99002) JOBNAME(WARWKLY)
```

Return Codes

See "Chapter 13. DFSMSrmm Return and Reason Codes" on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

CHANGEOWNER: Changing Owner Information

Use the CHANGEOWNER subcommand as shown in Figure 151 to update information about an owner or group of owners defined to DFSMSrmm by an owner ID. You must supply the owner ID for which you want to change information.

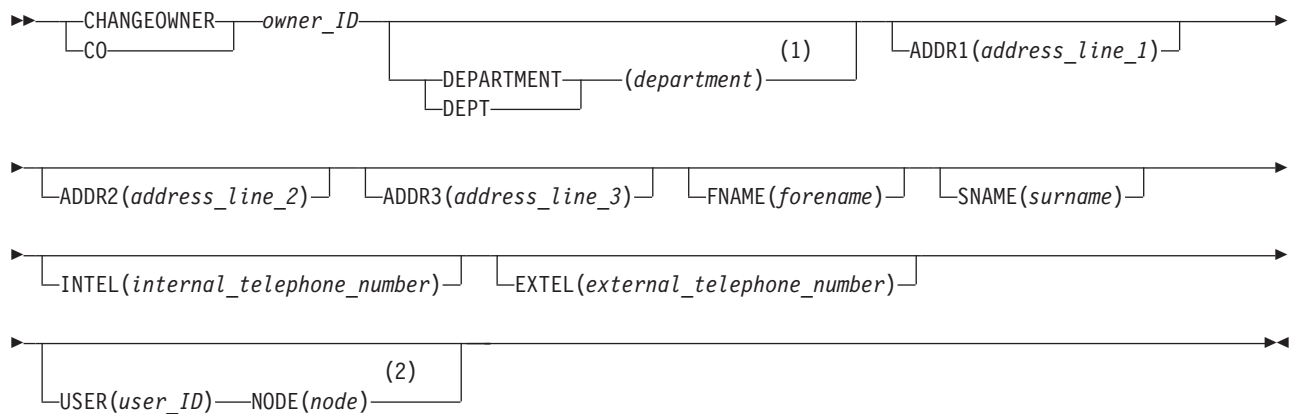
Example

Task: Change the following details for the owner whose owner ID is MAXWELL: new internal telephone number is 321 4567; user ID and node are ALYN at NODEUK.

Command:

```
RMM CHANGEOWNER MAXWELL INTEL('321 4567') -
  USER(ALYN) NODE(NODEUK)
```

Syntax



Notes:

1. Cannot be all blank characters
2. The values you specify for USER(user_ID) and NODE(node) work together; if you delete one, you must also delete the other.

Figure 151. CHANGEOWNER Syntax Diagram

Operands

ADDR1(address_line_1)

Specify to change the first address line of the owner's address. An address line is one to forty characters enclosed in single quotes if it contains any special characters or blanks.

ADDR2(address_line_2)

Specify to change the second address line of the owner's address. An address line is one to forty characters enclosed in single quotes if it contains any special characters or blanks.

ADDR3(address_line_3)

Specify to change the third address line of the owner's address. An address line is one to forty characters enclosed in single quotes if it contains any special characters or blanks.

CHANGEOWNER Subcommand

DEPARTMENT/DEPT(*department*)

Specify to change the name of the owner's department. A department name is one to forty characters and must not be all blanks. Enclose the department name in single quotes if it contains any special characters or blanks.

EXTEL(*external_telephone_number*)

Specify to change the owner's external telephone number. An external telephone number is one to twenty characters enclosed in single quotes if it contains any special characters or blanks.

FNAME(*forename*)

Specify to change the owner's forename, or first name. A forename is one to twenty characters enclosed in single quotes if it contains any special characters or blanks.

INTEL(*internal_telephone_number*)

Specify to change the owner's internal telephone number. An internal telephone number is one to eight characters enclosed in single quotes if it contains any special characters or blanks.

NODE(*node*)

Specify to change the owner's node name, to be used to notify the owner of any change in the status of owned volumes. A node name is one to eight alphanumeric characters or \$, #, @. Use both NODE() and USER() to clear an electronic address.

owner_ID

Specify to identify the ID of the owner for which you are changing information. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. This operand is required and must immediately follow the CHANGEOWNER subcommand.

SNAME(*surname*)

Specify to change the owner's surname, or last name. A surname is one to twenty characters enclosed in single quotes if it contains any special characters or blanks.

USER(*user_ID*)

Specify to change the owner's user ID, to be used to notify the owner of any change in the status of owned volumes. This can be different than the owner ID used to identify the owner. A user ID is one to eight alphanumeric characters. Use both USER() and NODE() to clear an electronic address.

Example

Task: Delete the user ID and node for the owner whose owner ID is AVISM.

Command:

```
RMM CHANGEOWNER AVISM USER() NODE()
```

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

CHANGEPRODUCT Subcommand

CHANGEPRODUCT: Changing Software Product Information

Use the CHANGEPRODUCT subcommand as shown in Figure 152 to update information about a software product defined to DFSMSrmm. You enter the software product number and, optionally, its version. If you do not supply the version, the default is V01R01M00, Version 1, Release 1, Modification Level 0.

Use the ADDVOLUME subcommand to associate a new volume with the software product. Use the CHANGEVOLUME subcommand to associate a volume that is already defined to DFSMSrmm but not associated with the software product. Use the DELETEVOLUME subcommand to disassociate a volume with a software product and release it. See “CHANGEVOLUME: Changing Volume Information” on page 282 and “DELETEVOLUME: Deleting Volume Information” on page 320 for more information.

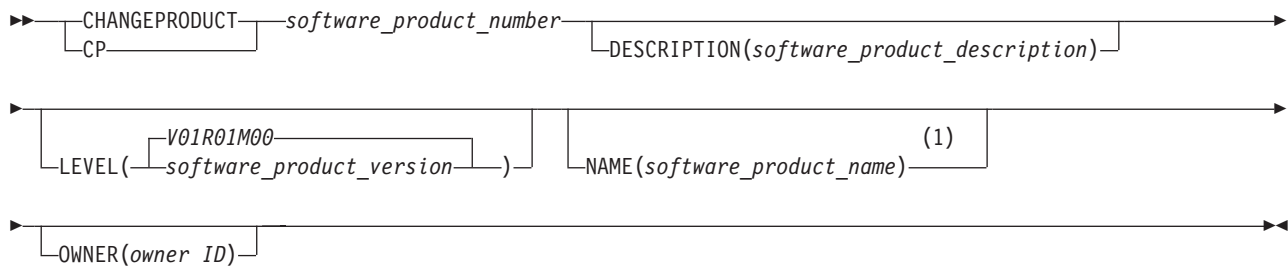
Example

Task: Change the owner of a software product with the product number, 5665-XA3. The new owner has the owner ID, CRUMPM.

Command:

```
RMM CHANGEPRODUCT '5665-XA3' OWNER(CRUMPM)
```

Syntax



Notes:

1. Cannot be all blank characters

Figure 152. CHANGEPRODUCT Syntax Diagram

Operands

DESCRIPTION(software_product_description)

Specify to change the description of the software product. Descriptive text can be one to thirty characters enclosed in single quotes if it contains any special characters or blanks.

LEVEL(software_product_version)

Specify to change the software product's version. Enter the version in the form, VnnRnnMnn, indicating the version, release, and modification level. 'nn' is two numbers in the range 00 to 99.

The default value is V01R01M00.

CHANGEPRODUCT Subcommand

NAME(*software_product_name*)

Specify to change the software product's name. A software product name is one to thirty characters enclosed in single quotes if it contains any special characters or blanks.

You can use the software product name to request lists of software products defined to DFSMSrmm. See "SEARCHPRODUCT: Creating a List of Software Products" on page 369 for more information.

OWNER(*owner*)

Specify to change the software product's designated owner. An owner ID is one to eight alphanumeric characters, or \$, #, or @; normally a RACF user ID or RACF group name. The first character must not be a number.

software_product_number

Specifies the number of the software product being changed. A software product number is one to eight characters enclosed in single quotes if it contains any special characters or blanks. This operand is required and must immediately follow the CHANGEPRODUCT subcommand.

Return Codes

See "Chapter 13. DFSMSrmm Return and Reason Codes" on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

CHANGEVOLUME Subcommand

CHANGEVOLUME: Changing Volume Information

Use the CHANGEVOLUME subcommand as shown in Figure 154 on page 287 to update details for a volume already defined to DFSMSrmm.

Provide a new rack number to change the volume's shelf location in the removable media library. The new rack number must be empty, or available for use. The old rack number is changed to EMPTY status. You also need to replace the external label on the volume to reflect the volume's new shelf location.

Supply a new owner ID to transfer ownership of the volume.

If you change the data set name, DFSMSrmm deletes all information about data sets on this volume and creates a new skeleton entry. Use the CHANGEDATASET subcommand to supply full data set details. See "CHANGEDATASET: Changing Data Set Information" on page 270 for more information.

Rules for Changing Volume Information: The rules for changing volume information are based on how the information is recorded, the status of the volume, if the request to change information is submitted by an authorized user, and where the volume resides.

Here is a summary of the rules for changing volume information using the RMM CHANGEVOLUME subcommand. See the CHANGEVOLUME syntax diagrams and operand descriptions for details about the volume information that can be changed.

Changing Non-restricted Volume Information: Non-restricted volume information as described in Table 25 is information that you can change when you own the volume or you have UPDATE access to STGADMIN.EDG.OWNER.userid.

Table 25. Non-restricted Volume Information

ACCESS	ACCOUNT	ADDUSERS	CONFIRMRELEASE
DELUSERS	DESCRIPTION	EXPDT	FEATCD
INITIALIZE	LEVEL	NUMBER	OWNER
OWNERACCESS	PREVVOL	RELEASEACTION	RETPD
SECLEVEL	SPECIALATTRIBUTES	STATUS	USE

Changing Volume Information Recorded by DFSMSrmm: If you do not manually define all information about a volume to DFSMSrmm, DFSMSrmm automatically records the remaining information when a data set on the volume is opened or closed. Table 26 lists the operands you can change if you are authorized to use the CHANGEVOLUME subcommand FORCE operand. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.

Table 26. Volume Information That Can Be Changed Under Special Conditions

ASDATE	ASTIME	COMPACTION	DENSITY
DSNAME	JOBNAME	LABEL	MEDIATYPE
NEWVOLUME	READDATE	RECORDINGFORMAT	WRITEDATE

Changing Volume Information Based on Volume Status: Table 27 on page 283 shows the operands that can only be used if the volume is in master or user status. If the volume is in scratch status, these operands are ignored unless you change the volume status. You can change the volume status by specifying STATUS (MASTER) or STATUS(USER).

CHANGEVOLUME Subcommand

Table 27. Volume Information for Volumes in Master and User Status

ACCESS	ACCOUNT	ADDUSERS	COMPACTION
DELUSERS	DESCRIPTION	DSNAME	EXPDT
FEATCD	JOBNAME	LEVEL	LOANLOC
NUMBER	OWNER	OWNERACCESS	PREVVOL
READDATE	RECORDINGFORMAT	RELEASEACTION	RETPD
SECLEVEL	SPECIALATTRIBUTES	STORGRP	WRITEDATE

You can use the operands shown in Table 28 for all volumes regardless of their status:

Table 28. Volume Information for All Volumes

ASDATE	ASTIME	DENSITY	EJECT
HOME	INIT	LABEL	LOCATION
MEDIATYPE	POOL	RACK	USE
STATUS			

Note: You cannot use the POOL and RACK operands if you are using the RMM CHANGEVOLUME subcommand to move volumes to system-managed libraries.

Changing Pending Actions: To reclaim volumes:

- From pending release, use the CHANGEVOLUME subcommand with a volume serial number and either the RETPD or EXPDT operands.
- From scratch status and recover the data on the volume, use the CHANGEVOLUME subcommand with either the STATUS(USER) or STATUS(MASTER) operands. You must have CONTROL access to STGADMIN.EDG.MASTER to reclaim a volume from scratch status.

You must have CONTROL access to the STGADMIN.EDG.MASTER resource to change the operands shown in Table 29.

Table 29. Volume Information for Pending Actions

CONFIRMMOVE	EJECT	HOME	LOCATION
LOANLOC	POOL	RACK	READDATE
STORAGEGROUP	WRITEDATE		

Changing Volume Location: If you want to move a volume to or from any location without performing vital record processing or storage location management processing, you can use the DFSMSrmm TSO CHANGEVOLUME subcommand to move it. DFSMSrmm updates the control data set to reflect the new location. Vital records processing and storage location processing will override changes made manually unless the volume is placed under manual move control. Be aware that during the next run of vital records processing or storage location processing, the volume might get marked for return back to the storage location from which it was moved.

When you use CHANGEVOLUME LOCATION(*library_name*), and the library name you specify is the name of an automated tape library dataserwer, the library name and volume information are checked in the TCDB. DFSMSrmm first checks the TCDB to see that the volume and library name are defined in the TCDB, then compares the location information in both the TCDB and the control data set. DFSMSrmm updates the control data set if there is a mismatch between information in the TCDB and control data set. If the volume is currently in a system-managed library, DFSMSrmm ejects the volume.

CHANGEVOLUME Subcommand

If the move destination for a volume is a manual tape library, DFSMSrmm requests that the volume is defined in the TCDB as residing in the named manual tape library.

Use the HOME operand to change the location where you want a volume returned when it is no longer retained by a vital record specification. When you use the LOCATION operand to change a volume's location and do not use HOME, the HOME location for the volume is set to the value specified for LOCATION.

Confirming Volume Moves and Release Actions: You must confirm volume movement and actions before DFSMSrmm can perform additional movement or actions you require. For example, if you have volumes that are ready to return to scratch status, you must confirm that the volumes have returned to their home location before DFSMSrmm can proceed with returning the volumes to scratch. Movement and action confirmation is one way that DFSMSrmm ensures that you are aware of the actions taken on your removable media.

You can confirm individual volume moves or actions using the RMM CHANGEVOLUME subcommand with a specific volume serial number. When you confirm an individual volume move or action, DFSMSrmm updates the control data set when the subcommand is processed.

You can also perform global confirmation of volume moves or actions. Global confirmation in DFSMSrmm is a two-step process.

1. You confirm that the volume has moved to its new location. Use the RMM CHANGEVOLUME subcommand or the DFSMSrmm ISPF dialog described in "Confirming Volume Movements to DFSMSrmm" on page 124 to confirm that the volume has moved. The status of the move or action changes from PENDING to CONFIRMED.
2. Run DFSMSrmm inventory management to update the DFSMSrmm control data set to reflect the completion of the move. See the *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information about running DFSMSrmm inventory management.

Figure 153 on page 286 shows the CONFIRMMOVE operand used for confirming volume movement. You can confirm that outstanding volume movements and actions for an individual volume or for all volumes with pending movement or action have taken place. To confirm an outstanding move or action for a volume, supply the volume serial number. Confirmation occurs when you use CHANGEVOLUME with the CONFIRM operand for a single volume.

To confirm an outstanding move or action for all volumes with outstanding moves or actions, perform a global confirmation by specifying an asterisk instead of a volume serial number.

If you want to undo global confirmation for volumes previously confirmed, you must do this before inventory management is run. Use the LISTCONTROL subcommand with the MOVES or ACTIONS operand to display the status of moves and actions. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information.

After checking the list of movement and actions, use CHANGEVOLUME with the NOCONFIRMMOVE or the NOCONFIRMRELEASE operand and a volume serial number of * to undo a previous global confirmation and to change the status of a move or action from CONFIRMED to PENDING.

CHANGEVOLUME Subcommand

Use the READYTOSCRATCH or NOTREADYTOSCRATCH operands to confirm moves for volumes that are ready to return to scratch, or those that are not, such as private volumes, or volumes with release actions other than just scratch pending.

Overriding Automatic Move Processing: Use the MANUALMOVE operand to manually control the movement of volumes. Use the AUTOMOVE operand to return to automatic processing. See “Using Manual Move Control” on page 117

To override automatic movement, you can put a volume under manual move control using the CHANGEVOLUME subcommand with the MANUALMOVE operand. The volume remains under manual move control until you issue the CHANGEVOLUME subcommand with the AUTOMOVE operand.

You can issue the CHANGEVOLUME subcommand with the LOCATION operand to set a destination for a volume if no destination is set for the volume. Note that the volume might move from the destination you set as a result of automatic processing. If you want the volume to stay in the destination you set, put the volume under manual move control.

Canceling Outstanding Volume Moves: You can also issue the CHANGEVOLUME subcommand with the LOCATION operand specifying the current location to cancel any outstanding move. This allows a new destination to be set by the CHANGEVOLUME subcommand or by automatic processing.

Changing Volume Serial Numbers: You can issue the CHANGEVOLUME subcommand with the NEWVOLUME operand to change the volume serial number for a volume defined to DFSMSrmm. This does not change the tape label on the physical volume but does allow you make changes to your inventory. If the volume serial number was recorded during O/C/EOV processing, you must also use the FORCE operand to change the volume serial number.

Example

Task: Delete the following user IDs from the access list of volume 8E1U02: OWNER12, OWNER25, OWNER44 and OWNER45. Keep the volume for another 365 days.

Command:

```
RMM CHANGEVOLUME 8E1U02 -  
  DELUSERS(OWNER12,OWNER25,OWNER44,OWNER45) RETPD(365)
```

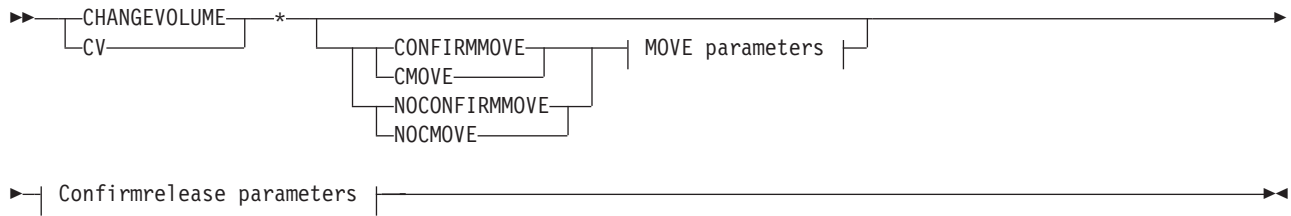
Task: Confirm all moves from location REMOTE to location SHELF which meet the Ready-to-Scratch criteria.

Command:

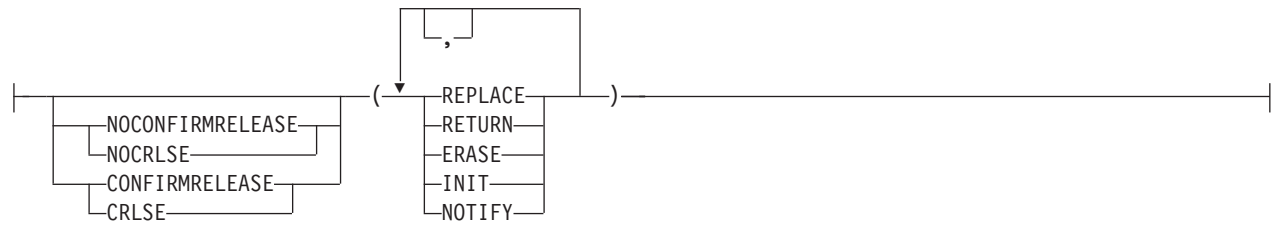
```
RMM CHANGEVOLUME * CONFIRMMOVE(REMOTE,SHELF,READYTOSCRATCH)
```

CHANGEVOLUME Subcommand

Syntax



Confirmrelease parameters:



MOVE parameters:

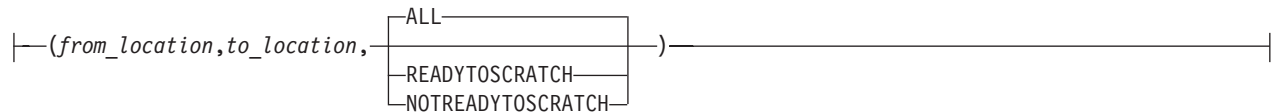
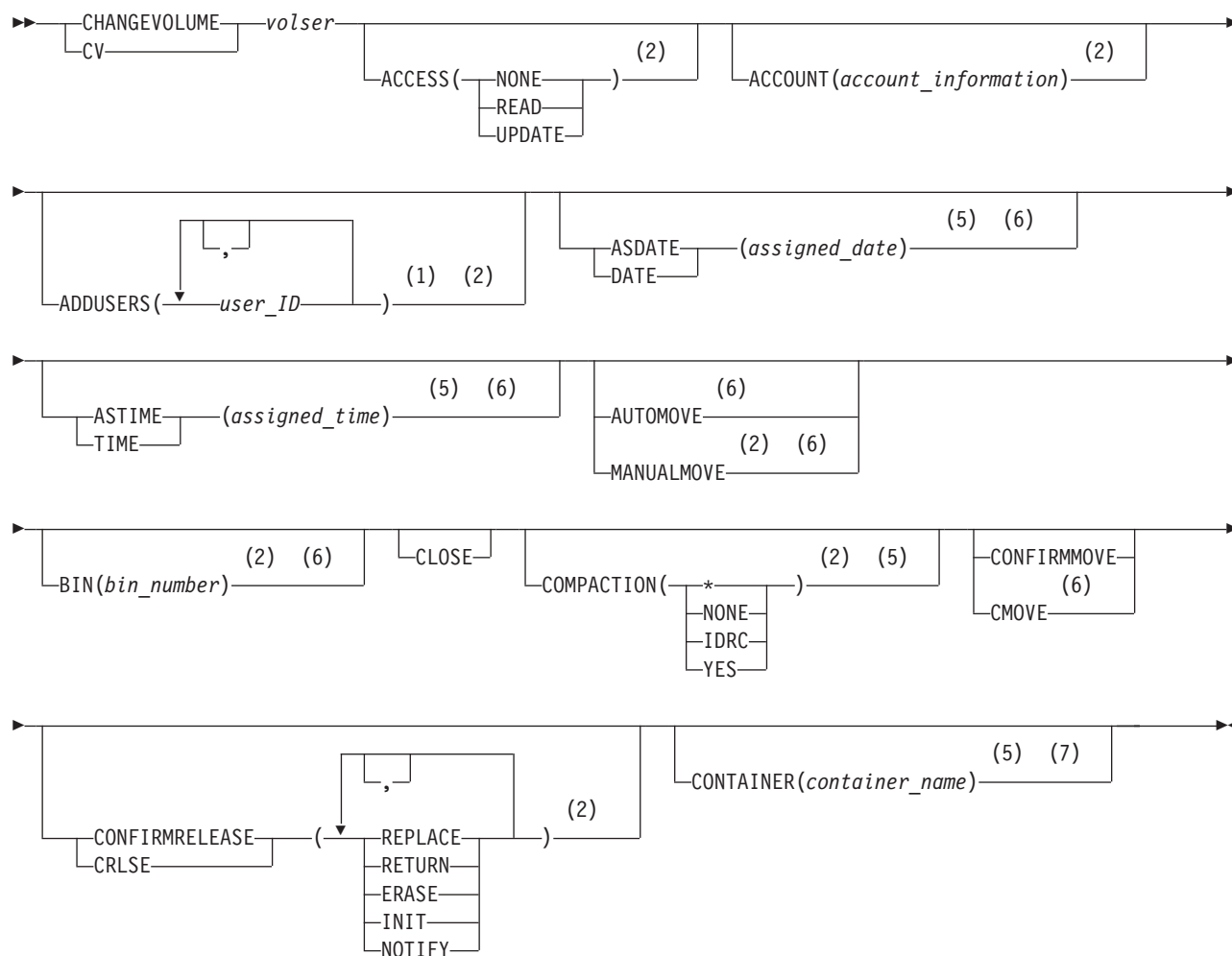


Figure 153. CHANGEVOLUME Syntax Diagram for Global Movement and Confirmrelease Parameters

Syntax



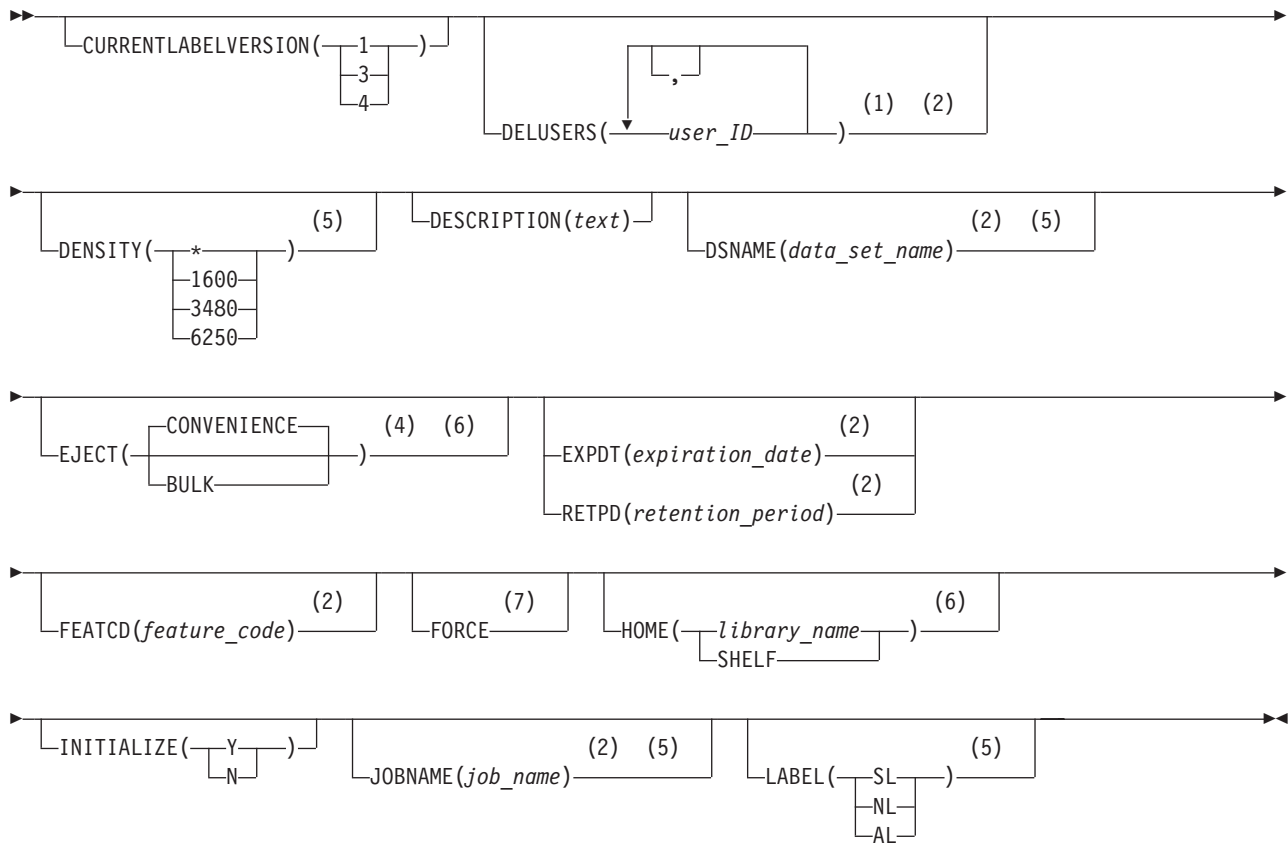
Notes:

1. You can specify a maximum of 12 user IDs.
2. Cannot be specified for a SCRATCH volume, unless you specify the STATUS parameter.
3. Cannot be specified if the volume resides in an IBM Tape Library Dataserver.
4. Can be specified only if the volume resides in a system-managed library.
5. Can be specified only if you also specify the FORCE operand. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.
6. Can be specified if the user has CONTROL access to STGADMIN.EDG.MASTER resource.
7. Import/Export support is available with APAR OW36342 or OW36343.

Figure 154. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters

CHANGEVOLUME Subcommand

Syntax

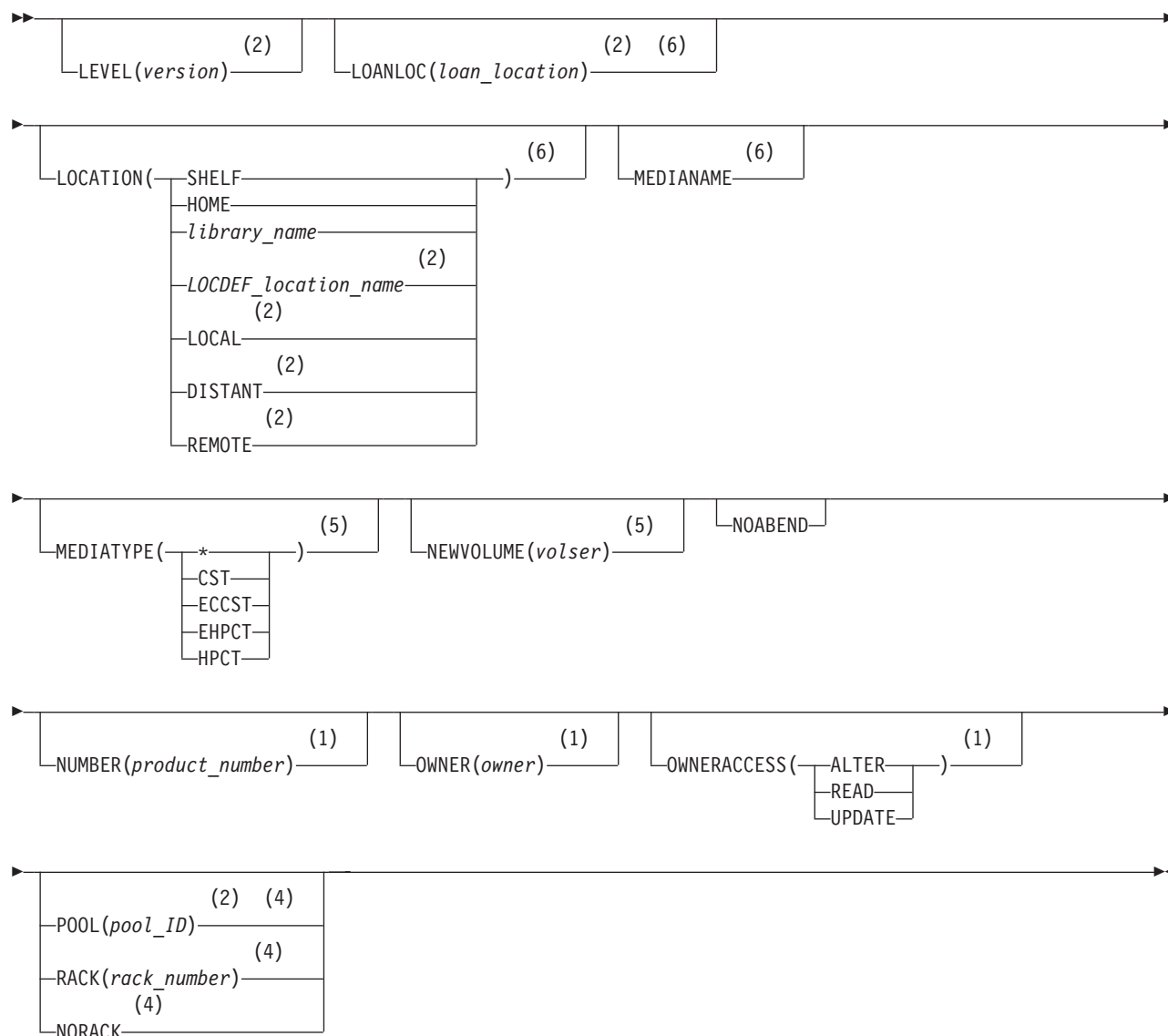


Notes:

1. You can specify a maximum of 12 user IDs.
2. Cannot be specified for a SCRATCH volume, unless you specify the STATUS parameter.
3. Cannot be specified if the volume resides in an IBM Tape Library Dataserver.
4. Can be specified only if the volume resides in a system-managed library.
5. Can be specified only if you also specify the FORCE operand. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.
6. Can be specified if the user has CONTROL access to STGADMIN.EDG.MASTER resource.
7. Can be specified to change a data set on a volume where information was recorded by DFSMSrmm during O/C/EOV processing. You must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource to use the FORCE operand.

Figure 155. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters (Cont.)

Syntax



Notes:

1. Cannot be specified for a SCRATCH volume, unless you specify the STATUS parameter.
2. Cannot be specified if the volume resides in an IBM Tape Library Dataserver.
3. Can be specified only if the volume resides in a system-managed library.
4. Can be specified if the user has CONTROL access to STGADMIN.EDG.MASTER resource.
5. Can be specified only if you also specify the FORCE operand. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.

Figure 156. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters (Cont.)

CHANGEVOLUME Subcommand

Syntax



Notes:

1. Cannot be specified for a SCRATCH volume, unless you specify the STATUS parameter.
2. Cannot be specified if the volume resides in an IBM Tape Library Dataserver.
3. Can be specified only if the volume resides in a system-managed library.
4. Can be specified if the user has CONTROL access to STGADMIN.EDG.MASTER resource.
5. Can be specified only if you also specify the FORCE operand. To use the FORCE operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.
6. Import/Export support is available with APAR OW36342 or OW36343.

Figure 157. CHANGEVOLUME Syntax Diagram for Specific Volume Parameters (Cont.)

Operands**ACCESS(NONE,READ,UPDATE)**

Specify to change the type of access to a volume for those users that are defined in the list of users (with the USERS operand). All users in the list have the same access level. DFSMSrmm uses the access information to build the RACF TAPEVOL access list.

The value can be one of the following:

NONE Users do not have access to the volume

READ Users have only read access

UPDATE

Users have write access to the volume

DFSMSrmm ignores this operand for scratch volumes if you do not specify either STATUS(MASTER) or STATUS(USER).

ACCOUNT(*account_information*)

Specify to change the accounting information recorded for the volume. Accounting information is one to forty characters enclosed in single quotes if it contains any special characters or blanks.

If you do not use this information, DFSMSrmm records it when it records information about the first data set on the volume. At that time, DFSMSrmm gets the accounting information from the account number of the job that created the first data set.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

ADDUSERS(*user_ID,user_ID,...*)

Specify to add to or change a list of user IDs and group names of users who can access a volume. Supply a maximum of twelve user IDs separated by blanks or commas. If you used RACF to maintain this access list, the RACF list is updated with your changes.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

ASDATE/DATE(*assigned_date*)

Specify to change the date when the volume was assigned to a user or returned to scratch status. Supply the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 93001.
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm uses 19 as the first two digits of the year. The dates 99365 and 99366 can be used only when the MAXRETPD NOLIMIT value is specified in parmlib. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER). This operand can be changed if you have authorization to use the FORCE operand.

CHANGEVOLUME Subcommand

ASTIME/TIME(*assigned_time*)

Specify to change the time a volume was assigned to a user or returned to scratch status. The format for ASTIME is: *hhmmss* where:

- *hh* is hours
- *mm* is minutes
- *ss* is seconds

For example, nine o'clock in the morning is 090000.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER). This operand can be changed if you have authorization to use the FORCE operand.

AUTOMOVE

Specify to return the volume to DFSMSrmm inventory management automatic movement control.

BIN(*bin_number*)

Use this operand to indicate a specific bin number to which the volume should be moved in a storage location. To assign a specific bin number, the volume's current location must be a shelf-managed storage location, or you must specify the LOCATION operand to identify a shelf-managed storage location that contains the bin number. If the storage location is installation defined DFSMSrmm uses the LOCDEF parameters to determine the media name to be used.

This operand is not valid for a scratch volume unless the status operand is also specified. The bin number you specify must be defined and empty or the command fails.

CLOSE

Specify to reset the volume OPEN condition.

COMPACTION(*,NONE,IDRC,YES)

Specifies the compaction technique used to record data on tape volumes. Use one of the following:

* The compaction is not known; or the volume is not a tape volume, and compaction does not apply. This is the default.

NONE No compaction was used to record data on the volume.

IDRC IDRC compaction, which DFSMSrmm displays as a compaction value of YES was used.

YES The data on the master or user tape volumes being added is compacted.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB. This operand can be changed if you have authorization to use the FORCE operand.

CONFIRMMOVE/CMOVE(*from_location,to_location*, ALL,

READYTOSCRATCH,NOTREADYTOSCRATCH)

Use the CONFIRMMOVE operand to confirm that a pending move for a volume has taken place.

CHANGEVOLUME Subcommand

To confirm a move for a single volume, use the CHANGEVOLUME subcommand with a volume serial number and the CONFIRMMOVE operand. You do not need to use any other CONFIRMMOVE values.

When performing a global confirm of volume movement, you can specify a *from_location*, *to_location* and ALL, READYTOSCRATCH, or NOTREADYTOSCRATCH. Use the following values for *from_location* and *to_location*:

ALL

All locations

library_name

A shelf location in a system-managed library. Library names must be one to eight alphanumeric characters, \$, @, or #, starting with a non-numeric character.

SHELF

A shelf location in a non-system-managed library

LOCAL

The local storage location

LOCDEF_location_name

To indicate that the volume should be stored in an installation defined storage location. *LOCDEF_location_name* can be any name up to eight characters long.

DISTANT

The distant storage location

REMOTE

The remote storage location

In addition to specifying the *from_location* and *to_location* with the CONFIRMMOVE operand, you can use ALL, READYTOSCRATCH, or NOTREADYTOSCRATCH as follows:

ALL

All volume moves. ALL is the default.

READYTOSCRATCH

A subset of all volume moves eligible to become scratch because no other release actions are pending

NOTREADYTOSCRATCH

These are private volumes or volumes with release actions pending other than return to scratch

To confirm that you have moved all volumes with the same outstanding move, use CONFIRMMOVE with *from* and *to* location values for the move, ALL, and a volume serial number of *.

You can supply location values as follows to identify which move you are confirming:

- **CONFIRMMOVE(*from_location*,*to_location*)** to confirm all moves from one location to another location
- **CONFIRMMOVE(*from_location*,ALL)** to confirm all moves from a location to any other locations

CHANGEVOLUME Subcommand

- **CONFIRMMOVE**(ALL,*to_location*) to confirm all moves to a location from any other location
- **CONFIRMMOVE**(ALL,ALL) to confirm all outstanding moves regardless of starting location and destination

You can use CONFIRMMOVE in conjunction with the LOCATION operand to have DFSMSrmm immediately record the volume's new location. You can set a new destination for a volume using the LOCATION operand as well.

When you move a volume to an automated tape library dataserer, the move is automatically confirmed as you enter the volume. If the system is down or DFSMSrmm is inactive, you can use the CHANGEVOLUME subcommand with CONFIRMMOVE to confirm the move if the volume has already been moved and now resides in that library.

To confirm a volume move between system-managed libraries, you must issue the CHANGEVOLUME subcommand with the CONFIRMMOVE operand from a system on which those libraries are defined. During DFSMSrmm inventory management, DFSMSrmm ignores the confirmation of volume moves between any libraries not defined on the system running inventory management, and defines those moves as pending.

See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for information on the LISTCONTROL subcommand, which you can use with the MOVES operand to display information about outstanding moves.

CONFIRMRELEASE/CRLSE(REPLACE,RETURN,ERASE,INIT,NOTIFY)

Specify to confirm that you have performed the requested actions and that release processing can continue. You only need to confirm those actions which you do out manually. DFSMSrmm confirms those actions it performs out automatically.

Use CONFIRMRELEASE with a volume serial to confirm an action for a volume. Use CONFIRMRELEASE with a volume serial of * to confirm that you have done out an action for all volumes awaiting the indicated action.

Use CONFIRMRELEASE with one of the following values to confirm actions taken manually for the volume being released. Use commas to separate these values from the first action you specify and from each other.

REPLACE

You have replaced the volume with a new volume

RETURN

You have returned the volume to its owner

ERASE

You have erased the volume

INIT

You have initialized the volume

NOTIFY

You have notified the owner of the volume's release.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

CONTAINER(*container_name*)¹²

Specifies the name of the location where a volume currently resides. The container can be a physical stacked volume. The value can be any alphanumeric or special characters up to 16 characters in length. To clear the in container value, specify CONTAINER(' ').

CURRENTLABELVERSION(1,3,or 4)

Specify to change the ISO/ANSI label version set for the volume.

There is no default.

DELUSERS

Specifies a list of user IDs and group names to be deleted from the list of users who can access a volume. Supply a maximum of twelve user IDs separated by blanks or commas.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

DENSITY(*,1600,3480,6250)

Specify to change the volume's recording density. For a 3420 tape reel this can be 1600 or 6250. For a 3480 tape cartridge use a value of 3480. Supply a value of * if you do not know the density. This operand can be changed if you have authorization to use the FORCE operand.

DESCRIPTION(*text*)

Specify to change the volume description. Descriptive text is one to thirty characters enclosed in single quotes if it contains any special characters or blanks.

DSNAME(*data_set_name*)

Specify to change the recorded name of the first data set on the volume. DFSMSrmm deletes all information previously recorded for the data set, and uncatalogs all the data sets on the volume. If the data set name you give matches the data set name on the volume, DFSMSrmm only uncatalogs subsequent data sets on the volume.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER). This operand can be changed if you have authorization to use the FORCE operand.

EJECT(CONVENIENCE,BULK)

Specifies that you want to eject the volume from an automated tape library or from a manual tape library dataserver. Use EJECT(BULK) to eject the volume to the high capacity output station. Use EJECT(CONVENIENCE) to eject the volume to the convenience output station.

If you use EJECT(BULK) for a volume residing in a manual tape library dataserver, DFSMSrmm handles this as a normal EJECT. Do not use this operand for a volume that is currently not residing in a system-managed library. CONVENIENCE is the default.

A logical volume cannot always be ejected. DFSMSrmm honors any request to eject a volume but the request can fail if the VTS does not support the request. Requests supported by a VTS include an eject for any volume in insert category or any volume in a category with the fast ready attribute. The latter is normally

12. Import/Export support is available with APAR OW36342 or OW36343.

CHANGEVOLUME Subcommand

limited to scratch volumes. You can only specify EJECT for logical volumes when they are scratch volumes, and then the volume serial number is removed from the inventory in the Library Manager, and from the TCDB.¹³

EXPDT(*expiration_date*)

Specify to change the volume expiration date. This is the date when the volume should be considered for release. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm defaults to the 20th century.

EXPDT is mutually exclusive with RETPD. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

FEATCD(*feature_code*)

Specify to change the software product feature code. A feature code is one to four alphanumeric characters.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

FORCE

Specify to change information that DFSMSrmm recorded during O/C/EOV processing. Using FORCE allows you to change information that DFSMSrmm recorded during O/C/EOV processing. You can also eject a volume out of a library even when the library is not known or offline by using the FORCE operand with the EJECT operand. When you eject a volume by using the FORCE operands and the EJECT operands, the TCDB is not updated. To use the FORCE operand with or without the EJECT operand, you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.

HOME(*library_name*,SHELF)

Specify to set the location where the volume should be returned to when no longer retained by a vital record specification. This can be either an automated or a manual tape library dataserer, or a shelf location in a non-system-managed library. Changing this value allows you to inform DFSMSrmm where the volume should be stored when it is no longer retained by a vital record specification; it does not cause any volume movement.

DFSMSrmm processes the HOME operand before the LOCATION operand.

You specify SHELF or a VTS library name for a logical volume.¹⁴

A home location name is one to eight alphanumeric characters starting with a non-numeric character.

13. Import/Export support is available with APAR OW36342 or OW36343.

14. Import/Export support is available with APAR OW36342 or OW36343.

INITIALIZE(Y/N)

Specify to change whether the volume should be initialized. Type Y to indicate that the volume requires initializing. You cannot use the volume until initialization is confirmed. Type N to indicate that the volume does not need to be initialized.

If you request initialization for a scratch volume, and the initialize action is still pending when you enter the volume into an automated tape library dataserer, DFSMSrmm defers initialization to DFSMSdfp labeling support. If the volume is later ejected without being initialized, DFSMSrmm reinstates the initialize action. If you use INIT(Y) for volumes residing in an automated tape library dataserer the action is set pending and you must initialize the volume before it can be used.

DFSMSrmm accepts the abbreviation INIT.

JOBNAME(*job_name*)

Specify to change the name of the job that created the first file on the volume. A job name is one to eight alphanumeric characters or \$, #, or @. The job name must start with an alphabetic character, \$, #, or @. You cannot use a generic jobname. Any jobname you use must be specific. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER). This operand can be changed if you have authorization to use the FORCE operand.

LABEL(SL,NL,AL)

Specifies the label type of the volume.

SL

Standard IBM volume label.

NL

No label

AL

ISO/ANSI labels

DFSMSrmm updates the label type when a data set on the volume is opened. This operand can be changed if you have authorization to use the FORCE operand.

LEVEL(*version*)

Specify to change the version of the software product with which the volume is associated. Supply version in the form, VnnRnnMnn, indicating the version, release and modification level. 'nn' is two numbers in the range 00 to 99. Use the ADDPRODUCT subcommand to add a new version of a software product number already defined to DFSMSrmm.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

LOANLOC(*loan_location*)

Specify to change the location where a volume resides other than the removable media library or a storage location. A loan location is up to eight characters enclosed in single quotes if it contains any special characters or blanks. For example, if you removed the volume from the removable media library, you could use your owner ID as the loan location to let others know where the volume is stored. You can clear the loan location by specifying a blank loan location.

CHANGEVOLUME Subcommand

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

LOCATION(SHELF,HOME,*library_name*,LOCDEF_location_name,built-in_name)

Specify to change a volume's location. When you give a location value, you manually request a volume move. You can specify any location.

If you do not use the HOME operand, LOCATION also sets the home location name in the DFSMSrmm control data set.

You can use the LOCATION operand to correct the location recorded by DFSMSrmm if it currently does not match the library name in the TCDB. This corrects the DFSMSrmm control data set, and no physical move is required.

Logical volumes are not ejected when LOCATION is used. When you change the location of a logical volume, DFSMSrmm processing cannot eject the volume from the VTS. If the location name you specify does not match the volume's current location, DFSMSrmm sets the destination if the new location is not shelf-managed. For volumes that are shelf-managed, DFSMSrmm sets the required location and volume movement is done during inventory management processing.

Use one of the following to request a volume move:

SHELF

To indicate that the volume be stored in a shelf location in a non-system-managed library

HOME

To indicate that the volume return from a storage location to the location identified by the HOME operand. This sets the home location name in the DFSMSrmm control data set.

library_name

To indicate that the volume be stored in a specific system-managed library.

This library can be either a manual tape library dataserver or an automated tape library. A library name is one to eight alphanumeric characters, or \$, #, or @, starting with a non-numeric character. If you change this value to the same location that DFSMSrmm has set for the volume, DFSMSrmm checks the TCDB to see if the volume is defined and in that library.

If the library is a manual tape library dataserver, DFSMSrmm adds the volume to the TCDB. If the library is an automated tape library dataserver and the volume does not currently reside in the automated tape library dataserver, DFSMSrmm sets a volume move in progress to get the volume moved to the automated tape library dataserver.

LOCDEF_location_name

To indicate that the volume should be stored in an installation defined storage location.

built-in_name

To indicate that the volume should be stored in an DFSMSrmm built-in storage location you can specify:

LOCAL

The LOCAL storage location

CHANGEVOLUME Subcommand

DISTANT

The DISTANT storage location

REMOTE

The REMOTE storage location

Use LOCATION with CONFIRMMOVE to mark the volume move as complete. You can use the LOCATION operand to cancel a pending move by specifying the volume's current location.

MANUALMOVE

Specify to turn off DFSMSrmm automatic move processing, or to cancel a pending move.

MEDIANAME

Specify this operand to change the media name for a volume so it matches the media name defined for it in the EDGRMMxx parmlib VLPOOL command.

MEDIATYPE(*,CST,ECCST,EHPCT,HPCT)

Specifies the volume's physical media type. Use one of the following:

* the volume is not a cartridge. This is the default.

CST Cartridge System Tape

ECCST

Enhanced Capacity Cartridge System Tape

EHPCT

Reserved for Extended High Performance Cartridge Tape

HPCT High Performance Cartridge Tape

If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB. This operand can be changed if you have authorization to use the FORCE operand.

NEWVOLUME(*volser*)

Specify this operand to change the volume serial number of a volume in the DFSMSrmm control data set. Note: Using this operand does not change the tape label on the physical media. DFSMSrmm associates the new volume serial number with the information that is recorded for the old volume.

DFSMSrmm assigns a rack number for the volume that matches the new volume serial number and changes the rack number that is associated with the old volume serial number to empty status. To change the rack number that DFSMSrmm assigned to the volume, use the RACK operands or POOL operands to assign a rack number to the volume. To keep the volume in its current shelf location, use the RACK operand with the rack number that was associated with the old volume serial number.

If you want to change a volume serial number that was recorded by DFSMSrmm during O/C/EOV processing, you must use the FORCE operand to force the change.

This is an optional operand which has no default.

NOABEND

Specify to reset the ABEND flag for a volume.

This is an optional operand which has no default.

CHANGEVOLUME Subcommand

NOCONFIRMMOVE/NOCMOVE(*from_location*,*to_location*,ALL)

Specify to reverse a previous move confirmation for one or more volumes. You can only reverse a previous move confirmation as long as inventory management has not started.

You must use NOCONFIRMMOVE with a volume serial of * and location values.

Any of the following values can be used for *from_location* and *to_location* to identify the moves for which you are reversing confirmation:

ALL

All locations

library_name

A shelf location in a system-managed library. Library names must be one to eight alphanumeric characters, or \$, #, or @, starting with a non-numeric character.

LOCDEF_location_name

To indicate that the volume should be stored in an installation defined storage location.

SHELF

A shelf location in a non-system-managed library

LOCAL

The LOCAL storage location

DISTANT

The DISTANT storage location

REMOTE

The REMOTE storage location

Supply location values as follows to identify the move confirmation you are reversing:

- **NOCONFIRMMOVE(*from_location*,*to_location*)** to reverse a move confirmation from one specific location to another location
- **NOCONFIRMMOVE(*from_location*,ALL)** to reverse a move confirmation from a location to any other locations
- **NOCONFIRMMOVE(ALL,*to_location*)** to reverse a move confirmation to a location from any other location.
- **NOCONFIRMMOVE(ALL,ALL)** to reverse a move confirmation for all volume moves.

You can only use ALL with the NOCONFIRMMOVE operand if you initially supplied it on the move confirmation you are reversing. The *from_location* and *to_location* values must also be exactly as specified on the initial confirmation with the CONFIRMMOVE operand. For example, to reverse a previous move confirmation initially supplied as

```
CHANGEVOLUME * CONFIRMMOVE(ALL,REMOTE)
```

you can use

```
CHANGEVOLUME * NOCONFIRMMOVE(ALL,REMOTE)
```

but you cannot use

```
CHANGEVOLUME * NOCONFIRMMOVE(ALL,ALL)
```

CHANGEVOLUME Subcommand

NOCONFIRMRELEASE/NOCRLSE(REPLACE,RETURN,ERASE,INIT,NOTIFY)

Specify to reverse a previous release action confirmation.

Use the CHANGEVOLUME subcommand with an asterisk to reverse a release action confirmation for all volumes awaiting the indicated action.

Use NOCONFIRMRELEASE with any of the following values to reverse confirmation of release actions performed manually for the volume being released. You can use one or more of the following operands separated from the first operand and from each other by commas:

REPLACE

You have not replaced the volumes with new volumes

RETURN

You have not returned the volumes to their owner

ERASE

You have not erased the volumes

INIT

You have not initialized the volumes

NOTIFY

You have not notified the owner that one or more owned volumes is eligible for release

You can only reverse a previous action confirmation as long as inventory management has not started.

NORACK¹⁵

Specify to remove an assigned rack number from any volume. This allows the volume to be entered as a logical volume.

NUMBER(*product_number*)

Specify to change the number of a software product with which the volume is associated. A software product number is one to eight characters enclosed in single quotes if it contains any special characters or blanks.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

OWNER(*owner*)

Specify to change the owner ID of the volume's owner. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. We suggest that you use a RACF user ID or RACF group name.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

OWNERACCESS(ALTER,READ,UPDATE)

Specify to change the owner's access to the volume. OWNERACCESS can be used with OWNER to define the initial RACF volume profile access. The value can be READ, UPDATE or ALTER.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

15. Import/Export support is available with APAR OW36342 or OW36343.

CHANGEVOLUME Subcommand

POOL(*pool_ID*)

Specify to change the pool ID of the pool where the volume is stored in the removable media library. A pool ID is one to five characters followed by an asterisk, and must be defined by your installation. If you supply a pool ID, do not give a rack number.

You cannot use this operand for a volume residing in an automated or manual tape library dataserwer. POOL is mutually exclusive with RACK.

PREVOL(*previous_volser*)

To add a single volume to the end of a multivolume chain, supply a fully qualified volume serial number which is one to six alphanumeric characters, \$, #, or @.

To remove a single volume from the end of a multivolume chain, specify PREVOL without any values as shown in the example on page 306.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

RACK(*rack_number*)

Specify a *rack_number* to change the shelf location of a volume in the removable media library. A full rack number is six alphanumeric characters. The rack number you use must be available for use. You cannot specify a rack number for a volume residing in a system-managed tape library. The volume serial number and the rack number must be the same for volumes residing in a system-managed tape library. RACK is mutually exclusive with POOL.

READDATE(*last_read_date*)

Specifies when the volume was last read. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm defaults to the 20th century.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER). This operand can be changed if you have authorization to use the FORCE operand.

RECORDINGFORMAT(*,18TRACK,36TRACK,128TRACK)

Specifies the basic recording format for tape volumes.

- * An asterisk indicates that the format is unknown or that the volume is not a tape volume. This is the default.

18TRACK

Data has been written to the volume in 18TRACKs.

36TRACK

Data has been written to the volume in 36TRACKs.

128TRACK

Data has been written to the volume in 128TRACKs. Recording format 128TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB. For scratch volumes, DFSMSrmm lets the system set this value when the volume is first used. This operand can be changed if you have authorization to use the FORCE operand.

RELEASEACTION(SCRATCH,REPLACE,RETURN,INIT,NOINIT,ERASE,NOERASE,NOTIFY,NONOTIFY)

Specify to change the action to be taken when the volume is eligible for release. Use RELEASEACTION with one or more values identifying the release actions to be taken. The first value describes what should happen to the volume at the time it is released. Any following values specify other actions to be performed for the released volume.

You can use one of the following mutually exclusive release actions:

SCRATCH

To indicate that the volume should be returned to scratch status. SCRATCH is mutually exclusive with RETURN.

REPLACE

To indicate that the volume should be replaced with a new volume and returned to scratch status.

RETURN

To indicate that the volume should be returned to its owner. RETURN is mutually exclusive with SCRATCH.

You can supply up to three of the following values, separated from any previous value and from each other by commas. Use NOINIT, NOERASE, and NONOTIFY to request that DFSMSrmm remove an existing release action of INIT, ERASE, and NOTIFY for the volume.

INIT

To request that DFSMSrmm initialize the volume

NOINIT

To specify that DFSMSrmm not initialize the volume

ERASE

To request that DFSMSrmm erase the volume

NOERASE

To specify that DFSMSrmm not erase the volume

NOTIFY

To request that DFSMSrmm notify the owner that the volume is being released

NONOTIFY

To specify that DFSMSrmm not notify the owner of the volume's release.

CHANGEVOLUME Subcommand

You can request multiple actions for a volume. For example, if you want an owner to be automatically notified when a volume is released, and if the volume is to be returned to scratch status and initialized, use the following:
RELEASEACTION(SCRATCH,INIT,NOTIFY)

RETPD is mutually exclusive with EXPDT. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

REQUIREDLABELVERSION(0,3,4)

Specify to change the ISO/ANSI label version in the VOL1 label for the volume when creating or rewriting a tape with AL type labels. Use 0 to clear the required label field.

There is no default.

RETPD(*retention_period*)

Specify the new volume retention period. This is the number of days for which DFSMSrmm retains the volume before considering it for release. The value is one to four decimal numbers and is added to today's date to compute the new expiration date. It cannot exceed the maximum retention period (MAXRETPD) set by your installation in parmlib member EDGRMMxx. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

SECLEVEL(*security_class*)

Specify to change a volume's security class. The value must be one to eight characters and must be one of the security classes defined for your installation.

Use the LISTCONTROL subcommand with the SECCLS operand to display the security classes defined for your location. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

SPECIALATTRIBUTES(NONE,RDCOMPAT)

Specifies any special attributes associated with the tape volume.

NONE

The tape volume has no special attributes.

RDCOMPAT

The tape volume was created using one format and can be mounted on a drive that supports reading but not writing of that format.

For example, a volume recorded at 18TRACK can be read by a device that writes at 36TRACK and also has the ability to read 18TRACK tape volumes.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

STATUS(MASTER,USER)

Specify to change the volume's status. The value can be one of the following:

MASTER

To indicate that the volume cannot be overwritten except when the data set names match.

CHANGEVOLUME Subcommand

USER

To indicate that the volume can be overwritten by any data set.

STORAGEGROUP/STORGRP(*storage_group_name*)

Specify to change the SMS-defined storage group to which the volume belongs. A storage group name is one to eight alphanumeric characters. DFSMSrmm validates this name against the current SMS configuration by ensuring that the storage group has been defined.

If you do not use a value, DFSMSrmm does not record a storage group name for the volume. If a volume is already defined in a TCDB with a different storage group name, you cannot use the CHANGEVOLUME command to change this name.

DFSMSrmm accepts the abbreviation STORGRP.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

TYPE(LOGICAL,PHYSICAL) ¹⁶

Specify to change the type that is defined to DFSMSrmm. The possible values are: LOGICAL which is a logical volume and PHYSICAL which is real, physical volume. TYPE(LOGICAL) is not allowed if the LOCATION specifies a system-managed library that is not a VTS or if the volume has a rack number that is different from the volume serial number. If the rack number matches the volume serial number, the rack number is removed from the volume and the rack is left empty. TYPE(PHYSICAL) is not allowed if the LOCATION specifies a system-managed library that is a VTS.

USE(MVS,VM)

Specify to change the operating system(s) where the volume can be used. The value(s) can be MVS and VM. You can select both by specifying MVS,VM with a comma as a separator.

volser,*

Volser specifies the serial number of the volume being changed. A volume serial number is one to six alphanumeric characters, or \$, #, or @, or special characters.

Specify an asterisk with the CONFIRMRELEASE or CONFIRMMOVE operand to confirm one or more outstanding actions or moves for volumes. Use an asterisk with the NOCONFIRMRELEASE or NOCONFIRMMOVE operand to reverse a confirmation for one or more outstanding actions or moves for volumes.

Either a volume serial number or an asterisk is required and must immediately follow the CHANGEVOLUME subcommand.

WRITEDATE(*last_write_date*)

Specify to change when the volume was last written to. Supply the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.

16. Import/Export support is available with APAR OW36342 or OW36343.

CHANGEVOLUME Subcommand

- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm defaults to the 20th century.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER). This operand can be changed if you have authorization to use the FORCE operand.

Examples

Task: Indicate that you want to confirm that all outstanding volumes waiting to be replaced have been replaced.

Command:

```
RMM CHANGEVOLUME * CONFIRMRELEASE(REPLACE)
```

Task: Request that DFSMSrmm retain volume 8E1U02, which is ready to be considered for release and returned to scratch status, for another five days.

Command:

```
RMM CHANGEVOLUME 8E1U02 RETPD(5)
```

Task: Update the location information for volume ABC123 to indicate that you are moving it to an automated tape library dataserer named MYATL.

Command:

```
RMM CHANGEVOLUME ABC123 LOC(MYATL)
```

Then enter the volume in the library.

Task: Add an existing volume ABC123 to a product.

Command:

```
RMM CHANGEVOLUME ABC123 NUMBER(1234-456) -  
FEATCD(1234) LEVEL(V01R01M00)
```

Task: If you use ISMF or the MVS console command LIBRARY EJECT to eject a volume from an automated tape library dataserer, DFSMSrmm marks the volume as being 'in transit' but does not set a destination. Mark the volume as no longer in transit.

Command:

```
RMM CHANGEVOLUME ABC123 LOCATION(new_destination) CONFIRMMOVE
```

Task: Remove the last volume in a multivolume chain.

Command:

```
RMM CHANGEVOLUME ABC123 PREVVOL
```

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEDBIN Subcommand

DELETEDBIN: Deleting Bin Number Information

The DELETEDBIN subcommand is an alias for the DELETEDRACK subcommand. See “DELETEDRACK: Deleting Shelf Location Information” on page 317 for the combined description of the DELETEDRACK and DELETEDBIN operands.

DFSMSRmm defines shelf space in storage locations as bin numbers. Use the DELETEDBIN subcommand as shown in Figure 158 on page 309 to delete information about shelf locations you are no longer using in built-in or installation defined storage locations.

To delete bin numbers from a built-in storage location, specify an asterisk as the bin number and the name of a built-in storage location, LOCAL, REMOTE, or DISTANT. Specify a COUNT value to indicate how many bin numbers DFSMSRmm deletes. DFSMSRmm deletes bin numbers starting with the highest bin number defined for the storage location and works down until it reaches either the count value you specify or a bin number still containing a volume. If DFSMSRmm cannot delete all the bin numbers you requested, it displays an error message indicating that it could not reach the count.

To delete one or more bin numbers from an installation defined storage location, provide the initial bin number, the location name, and the media name. Specify a COUNT value to indicate how many bin numbers DFSMSRmm deletes. DFSMSRmm deletes bin numbers starting from the initial bin number and deletes bin numbers following the initial bin number until it reaches the count value you specify or a bin number containing a volume.

You can use location names and media names on the DELETEDBIN subcommand that are not currently defined to DFSMSRmm. This lets you clean up bin numbers that have been defined in error or after the change or removal of LOCDEF commands from the EDGRMMxx parmlib.

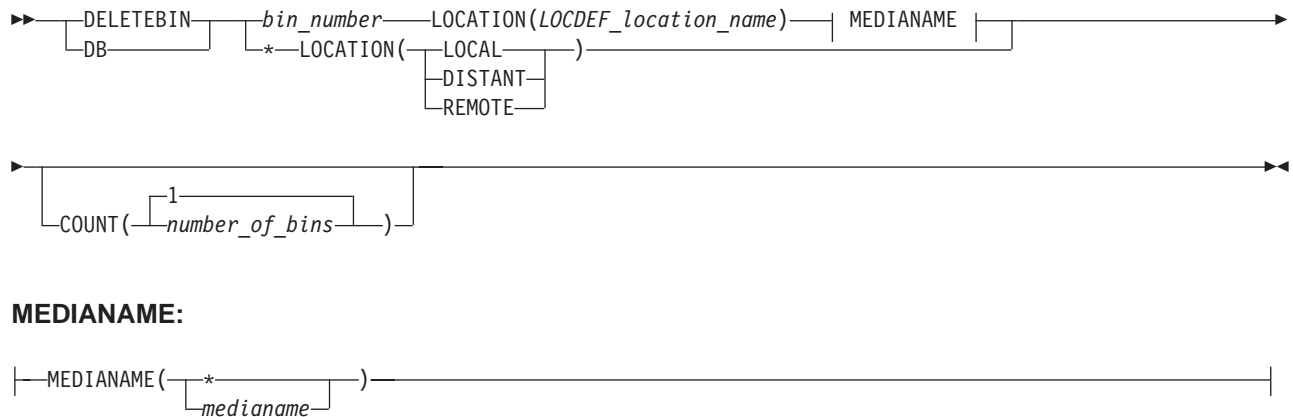
Example

Task: Remove information about eight shelf locations in the installation defined storage location MYLOC that no longer have volumes assigned to them. The bin numbers are A00001 to A00008 inclusive.

Command:

```
RMM DELETEDBIN A00001 LOCATION(MYLOC) COUNT(8) MEDIANAME(SQUARE)
```

Syntax



MEDIANAME:

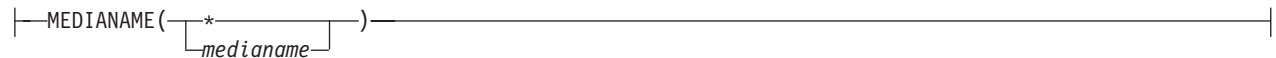


Figure 158. DELETEDBIN Syntax Diagram

Operands

bin_number,*

Immediately following the DELETEDBIN subcommand you must use either a bin number for an installation defined storage location or an * for a built-in storage location. You must also give a location name and media name.

A bin number in an installation defined storage location is six alphanumeric or national characters in any combination.

Specify an * to delete a bin number from a built-in storage location. When you use one of the built-in storage location names, LOCAL, DISTANT, or REMOTE, you do not provide a bin number because DFSMSrmm keeps track of bin numbers for built-in storage locations. The bin_number to be deleted must be empty.

COUNT(*number_of_bins*)

Specifies how many bin numbers DFSMSrmm deletes from a specified storage location. The value is one to five numbers. The maximum allowable decimal value is 99999.

The default value is 1.

LOCATION(LOCAL,DISTANT,REMOTE,LOCDEF_location_name)

Specifies the storage location where you want to delete shelf space.

Installation defined storage locations are names up to eight characters long and are defined using the LOCDEF parmlib command. You can delete bin numbers in installation defined storage locations even if the LOCDEF command for the installation is not in the current parmlib. To delete one or more bin numbers from an installation defined storage location, provide an initial bin number. MEDIANAME must also be specified. You can specify a COUNT value. If you use one of the built-in storage location names, LOCAL, DISTANT, or REMOTE, use an * as the bin number. You can specify a COUNT value. You cannot use MEDIANAME with a built-in storage location name.

MEDIANAME(*medianame*,*)

Defines the media that can reside in a shelf location. *medianame* can be eight characters. Specify an * to delete bin numbers from a built-in storage location.

DELETEDBIN Subcommand

Example

Task: Remove ten empty bin numbers from the DISTANT storage location. If there are 100 bin numbers defined in the storage location, DFSMSRmm deletes bin numbers 91 through 100 inclusive.

Command:

```
RMM DELETEDBIN * LOCATION(DISTANT) COUNT(10)
```

Task:

If you changed DFSMSRmm built-in storage locations, LOCAL, DISTANT and REMOTE, to installation defined locations, you might still have some DFSMSRmm assigned bin numbers for the location. To delete these bin numbers, issue the DELETEDBIN subcommand with an * for the bin number and provide one of the built-in storage location names. Use the RMM LISTCONTROL subcommand to find out how many bin numbers can be deleted.

Command:

```
RMM DELETEDBIN * LOCATION(REMOTE) COUNT(nnnnn)
```

Return Codes

See “Chapter 13. DFSMSRmm Return and Reason Codes” on page 401 for DFSMSRmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSRmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSRmm sets a reason code.
16	Error. The DFSMSRmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEDATASET: Deleting Data Set Information

Use the DELETEDATASET subcommand as shown in Figure 159 to delete information recorded by DFSMSrmm about a data set. DFSMSrmm also uncatalogs the data set, and any other data sets recorded on the same volume with higher data set sequence numbers. You specify the data set name and the serial number of the volume where the data set resides.

You must also specify a sequence number if the data set for which you are deleting information is not the first data set on the volume. If you do not specify a sequence number, DFSMSrmm assumes the data set is the first file on the volume. Unless the data set for which you are deleting information is the last data set on the volume, DFSMSrmm deletes information about all subsequent data sets on the volume.

Note: DFSMSrmm will not allow you to delete any data set on a volume if DFSMSrmm recorded data set information during O/C/EOV processing.

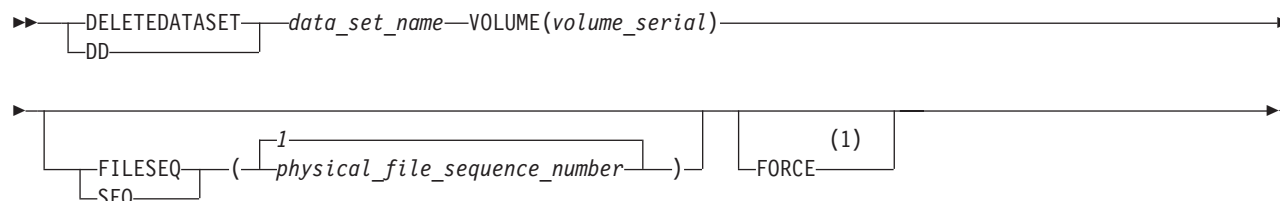
Example

Task: Delete information about all data sets residing on volume 8E1U01. The data set PREFIX.MYDATA.DATA is the first data set on the volume.

Command:

```
RMM DELETEDATASET 'PREFIX.MYDATA.DATA' VOLUME(8E1U01)
```

Syntax



Notes:

1. Specify to delete a data set from a volume where information was recorded by DFSMSrmm during O/C/EOV processing. You must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource to use the FORCE operand.

Figure 159. DELETEDATASET Syntax Diagram

Operands

data_set_name

Specifies the name of the data set for which you want to delete information. The name follows standard MVS naming conventions for data sets and must not include a member name. This operand is required and must immediately follow the DELETEDATASET subcommand.

FILESEQ/SEQ(1,physical_file_sequence_number)

Specifies the relative position of the data set on the volume. The minimum allowable decimal value is 1. The maximum allowable decimal value is 9999.

DELETEDATASET Subcommand

The default value is 1.

FORCE

Specify to override the restriction that information that DFSMSrmm recorded during O/C/EOV processing cannot be changed. Using FORCE allows you to delete data set information that DFSMSrmm recorded during O/C/EOV processing. To use this operand you must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource.

VOLUME(*volume_serial*)

Specifies the serial number of the volume where the data set resides. A volume serial number is one to six alphanumeric characters, or \$, #, or @, or special characters. A volume serial number is required.

Example

Task: Delete information about a data set named PREFIX.MYDATA.DATA that is the fourth data set on volume 8E1U01, and also delete information about all subsequent data sets on the volume.

Command:

```
RMM DELETEDATASET 'PREFIX.MYDATA.DATA' VOLUME(8E1U01) SEQ(4)
```

If prefix is your own TSO PROFILE PREFIX, you can enter:

```
RMM DELETEDATASET MYDATA.DATA VOLUME(8E1U01) SEQ(4)
```

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEOWNER: Deleting Owner Information

Use the DELETEOWNER subcommand as shown in Figure 160 to delete information about an owner defined to DFSMSrmm. You must specify the name or owner ID of the owner to be deleted.

When you delete the owner ID, you can optionally transfer ownership of any owned volumes to another owner already defined to DFSMSrmm. If you do not want to transfer ownership, use the DELETEVOLUME subcommand to release any owned volumes before you delete information about the owner. See “DELETEVOLUME: Deleting Volume Information” on page 320 for more information.

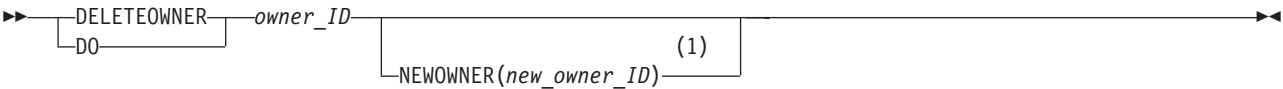
Example

Task: Delete the owner ID DARREN and transfer ownership of volumes owned by DARREN to the new owner, WOODSTER.

Command:

```
RMM DELETEOWNER DARREN NEWOWNER(WOODSTER)
```

Syntax



Notes:

1. Must be specified if the owner owns one or more volumes.

Figure 160. DELETEOWNER Syntax Diagram

Operands

NEWOWNER(new_owner_ID)

Specify the owner ID of the new owner to whom DFSMSrmm transfers volume ownership. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. We suggest that you use a RACF user ID or RACF group name. The new owner you specify must already be defined to DFSMSrmm. This operand is required if the owner being deleted owns volumes.

owner_ID

Specify the owner ID to be deleted from the DFSMSrmm control data set. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. This operand is required and must immediately follow the DELETEOWNER subcommand.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

DELETEOWNER Subcommand

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEPRODUCT: Deleting Software Product Information

Use the DELETEPRODUCT subcommand as shown in Figure 161 to delete information about a software product defined to DFSMSrmm. You can also release all volumes associated with the software product version you specify. You specify the software product number and, optionally, its version. If you do not specify the version, the default is V01R01M00, Version 1, Release 1, Modification Level 0.

Example

Task: Delete information about software product, 5665-XA3, Version V03R03M01, and release volumes associated with it.

Command:

```
RMM DELETEPRODUCT '5665-XA3' LEVEL(V03R03M01) RELEASE
```

Syntax

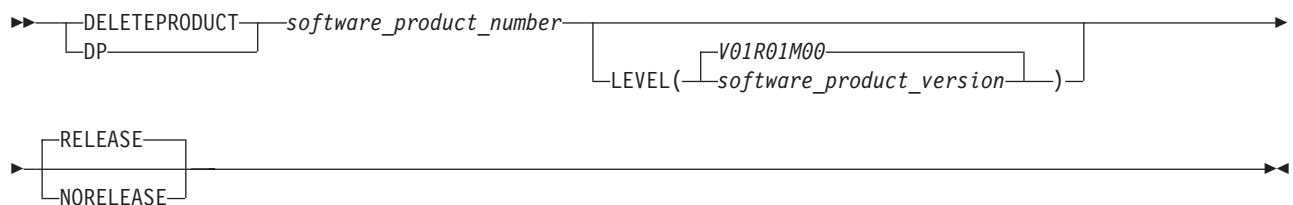


Figure 161. DELETEPRODUCT Syntax Diagram

Operands

LEVEL(*software_product_version*)

Specifies the version of the software product for which you are deleting information. Specify the version in the form, VnnRnnMnn, indicating the version, release, and modification level. 'nn' is two numbers in the range 00 to 99.

The default value is V01R01M00.

NORELEASE

Specify to retain all volumes associated with the specified software product version.

software_product_number

Specifies the number of the software product for which you are deleting information. A software product number is one to eight characters enclosed in single quotes if it contains any special characters or blanks. This operand is required and must immediately follow the DELETEPRODUCT subcommand.

RELEASE

Specify to release all volumes associated with the software product version.

RELEASE is the default value.

Return Codes

See "Chapter 13. DFSMSrmm Return and Reason Codes" on page 401 for DFSMSrmm reason codes.

DELETEPRODUCT Subcommand

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETERACK: Deleting Shelf Location Information

This section describes the combined description for the DELETERACK subcommand and its alias DELETEDBIN. See “DELETEDBIN: Deleting Bin Number Information” on page 308 for information about using the RMM DELETEDBIN subcommand alias.

Use the DELETERACK subcommand as shown in Figure 162 on page 318 to delete information about shelf locations you are no longer using in your removable media library or your storage locations. DFSMSrmm defines shelf space in the removable media library as rack numbers and bin numbers in storage locations.

See “DELETEDBIN: Deleting Bin Number Information” on page 308 for information about using the RMM DELETERACK subcommand alias.

If you are deleting a specific rack number from the removable media library, you must specify a six-character rack number. If you are deleting more than one rack number, you must specify the initial rack number and a COUNT value. To delete rack numbers, DFSMSrmm uses the current VLPOOL definitions to determine the media name of the rack numbers. You can use the DELETERACK subcommand with the MEDIANAME operand to delete empty rack numbers where the media name no longer matches to the VLPOOL media name.

To delete bin numbers from a built-in storage location, specify an asterisk as the rack number and the name of a storage location. Specify a COUNT value to indicate how many bin numbers DFSMSrmm deletes. DFSMSrmm deletes bin numbers starting with the highest bin number defined for the storage location and works down until it reaches either the count value you specify or a bin number containing a volume. If DFSMSrmm cannot delete all the bin numbers you requested, it displays an error message indicating that it could not reach the count. To delete bin numbers from an installation defined storage location, provide an initial bin number, a storage location name, and a media name. You can also provide the number of bins you want to delete. DFSMSrmm deletes bin numbers starting from the initial bin number and deletes bin numbers following the initial bin number until it reaches the count value you specify or a bin number containing a volume.

Example

Task: Remove information about ten shelf locations in the removable media library that no longer have volumes assigned to them. The rack numbers are A37652 to A37661 inclusive.

Command:

```
RMM DELETERACK A37652 COUNT(10)
```

DELETERACK Subcommand

Syntax

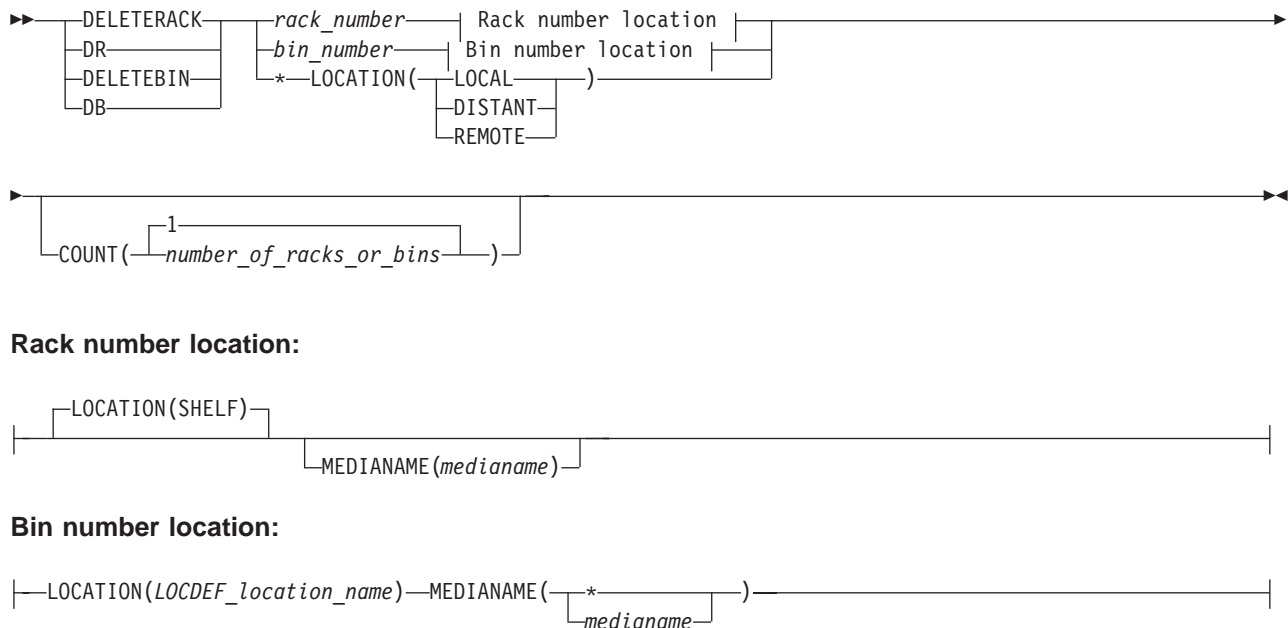


Figure 162. DELETERACK Syntax Diagram

Operands

*bin_number,**

Specify a bin number to delete a shelf location in an installation defined storage location. A bin number in an installation defined storage location is six alphanumeric or national characters in any combination. You must also give a location name and media name. The bin number you want to delete must be empty.

Immediately following the DELETEDBIN subcommand you must use either a bin number for an installation defined storage location or an * for a built-in storage location.

Specify an * to delete a bin number from a built-in storage location. If you use one of the built-in storage location names, LOCAL, DISTANT or REMOTE, DFSMSRmm assigns bin numbers.

COUNT(*number_of_racks_or_bins*)

Specifies how many rack numbers DFSMSRmm deletes from the removable media library, or how many bin numbers for a specified storage location. The value is one to five numbers. The maximum allowable decimal value is 99999.

The default value is 1.

LOCATION(SHELF,LOCAL,DISTANT,REMOTE,LOCDEF_location_name)

Specifies the location from which you want to delete shelf space. Use SHELF to delete shelf locations from your removable media library. LOCAL, DISTANT, and REMOTE are the DFSMSRmm built-in storage location names. You cannot use MEDIANAME with a built-in storage location name. LOCDEF_defined_name is any eight character installation defined storage location name. To delete bin numbers from an installation defined storage location, you provide the bin

DELETERACK Subcommand

numbers to delete. MEDIANAME must also be specified. If you do not use the LOCATION operand, DFSMSrmm deletes rack numbers from the removable media library.

MEDIANAME(*medianame*,*)

Defines the media name for the bin numbers or rack numbers to be deleted when the existing DFSMSrmm EDGRMMxx parmlib VLPOOL media names does not match. For bin numbers, any media name or * can be used. For rack numbers, you only need to specify this operand when the existing DFSMSrmm EDGRMMxx parmlib VLPOOL media name does not match the rack number media name.

rack_number

Specifies a single rack number to be deleted, or an initial rack number if you are deleting more than one rack number from the removable media library. A rack number is six alphanumeric or national characters. You cannot specify a generic rack number. The bin number you want to delete must be empty.

A rack number, bin number or an asterisk is required and must immediately follow the DELETERACK or DELETEDBIN subcommand.

Example

Task: Remove ten empty bin numbers from the DISTANT storage location.

Command:

```
RMM DELETEDBIN * LOCATION(DISTANT) COUNT(10)
```

If there are 100 bin numbers defined in the storage location, DFSMSrmm deletes bin numbers 91 through 100 inclusive.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEVOLUME Subcommand

DELETEVOLUME: Deleting Volume Information

Use the DELETEVOLUME subcommand as shown in Figure 163 on page 321 to delete information recorded by DFSMSrmm about a volume or to manually release a volume. You can release a volume anytime before data sets on the volume expire, or before the end of any retention period set by one or more vital record specifications. You must specify a volume serial number.

Use one of three operands to either schedule the volume for reuse, or to delete information about a volume:

RELEASE

To use the release actions specified for the volume and schedule it for reuse

REMOVE

To delete information about a scratch volume

FORCE

To delete information about any volume

When you use RELEASE, DFSMSrmm begins processing any release actions specified for the volume. If you release a volume currently residing in a storage location or in transit between locations, DFSMSrmm indicates that the volume is in release pending status and waits until the volume returns to the removable media library before scheduling any release actions specified for it.

To use the DELETEVOLUME RELEASE subcommand, access to the RACF profile STGADMIN.EDG.RELEASE is required when the profile is defined. If the RACF profile STGADMIN.EDG.RELEASE is not defined, READ access to the RACF profile STGADMIN.EDG.MASTER is required. See *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information.

If you use either the FORCE or REMOVE operands to delete information about a volume, DFSMSrmm deletes information about any data sets on the volume. If the volume is in a system-managed tape library, you can also specify the EJECT operand to direct the volume to an exit station. For volumes residing in an automated tape library dataserver, the default is to eject the volume to a convenience output station. You can also specify EJECT(BULK) to eject the volume to a high capacity output station. DFSMSrmm ignores the EJECT operand if you do not specify either FORCE or REMOVE. Specify NOEJECT with either FORCE or REMOVE if you do not want to direct the volume to an exit station.

DFSMSrmm ignores the EJECT operand if you specify it for a volume that is not currently residing in a system-managed tape library or for a logical volume in private status.. For a volume in scratch status, an eject is issued and the VTS Library Manager database entry for the volume might be purged.¹⁷

When you use FORCE or REMOVE for a volume in a system-managed tape library, DFSMSrmm purges any information about the volume from the DFSMSrmm control data set. Any TCDB information is purged once the volume is successfully ejected. This also applies if the volume currently resides in a non-system-managed tape library but is considered part of a system-managed tape library and is defined in the TCDB as shelf resident.

See “Releasing Volumes” on page 148 for more information on releasing volumes.

17. Import/Export support is available with APAR OW36342 or OW36343.

Example

Task: Delete information about the scratch volume with volume serial number 8E1U01.

Command:

```
RMM DELETEVOLUME 8E1U01 REMOVE
```

Syntax

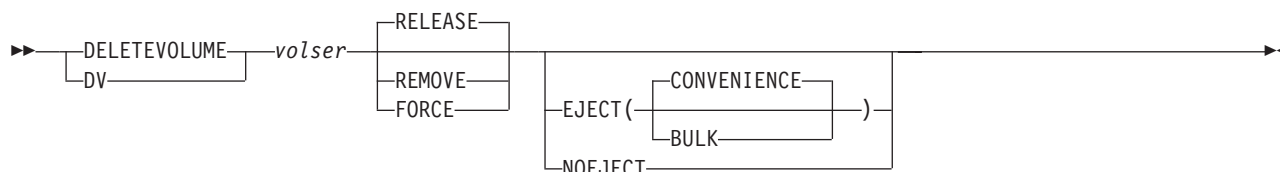


Figure 163. DELETEVOLUME Syntax Diagram

Operands

EJECT(CONVENIENCE,BULK)

Specify to direct a volume to an exit station, if you have also specified either the FORCE or the REMOVE operands. Specify EJECT(BULK) to eject the volume to the high capacity output station. Specify EJECT(CONVENIENCE) to eject the volume to the convenience output station.

If you specify EJECT for a volume residing in a system-managed tape library, information about the volume is purged from the TCDB once the volume has been successfully ejected, and it is deleted from the DFSMSrmm control data set. If you specify EJECT for a logical volume in SCRATCH status, DFSMSrmm deletes the volume but does not perform an EJECT to the convenience station.

If you specify EJECT for a volume that is currently not in a system-managed library, DFSMSrmm ignores the EJECT. However, if the volume is considered part of the system-managed tape library and is defined in the TCDB as shelf resident, information about the volume is purged from the TCDB and from the DFSMSrmm control data set.

When specifying FORCE or REMOVE for a volume in a system-managed library, the volume is ejected to the convenience output station if you do not specify EJECT. CONVENIENCE is the default.

FORCE

Specify to delete all information about a volume regardless of its current status, and to change the status of the rack number or bin number associated with the volume to empty. DFSMSrmm also uncatalogs all data sets on the volume.

If the volume resides in a system-managed library, DFSMSrmm ejects the volume and deletes the volume information in the DFSMSrmm control data set. Any volume information in the TCDB is purged once the volume is successfully ejected. If the volume resides in a non-system-managed library, DFSMSrmm simply deletes information from the DFSMSrmm control data set, unless the volume is considered part of a system-managed tape library and is defined in the TCDB as shelf resident.

DELETEVOLUME Subcommand

You can specify the EJECT operand with FORCE to direct the volume to an exit station other than the convenience output station. During error recovery, you can use NOEJECT with FORCE to prevent the volume from being ejected.

FORCE is mutually exclusive with RELEASE and REMOVE. RELEASE is the default.

NOEJECT

Specify to prevent the volume from being ejected, if you also specified either the FORCE or REMOVE operands. For example, you can specify DELETEVOLUME FORCE NOEJECT to purge information about the volume from the DFSMSrmm control data set without ejecting the volume. Volume information in the TCDB remains unchanged. You can use this function during error recovery processing.

RELEASE

Specify to release the volume according to the release actions set for the volume.

RELEASE is mutually exclusive with FORCE and REMOVE. RELEASE is the default.

REMOVE

Specify to delete information recorded by DFSMSrmm about a scratch volume that is no longer wanted, and to change the status of the rack number associated with the volume to empty. You can only specify REMOVE for a scratch volume.

If the volume resides in a system-managed library, DFSMSrmm ejects the volume and deletes the volume information in the DFSMSrmm control data set. Any volume information in the TCDB is purged once the volume is successfully ejected. If the volume resides in a non-system-managed library, DFSMSrmm simply deletes information from the DFSMSrmm control data set, unless the volume is considered part of a system-managed tape library and is defined in the TCDB as shelf resident.

You can specify the EJECT operand with REMOVE to direct the scratch volume to an exit station other than the convenience output station. During error recovery, you can use NOEJECT with REMOVE to prevent the volume from being ejected.

REMOVE is mutually exclusive with FORCE and RELEASE. RELEASE is the default.

volser

Specifies the volume serial number of the volume to be deleted. A full volume serial number is one to six alphanumeric characters, \$, #, or @, or special characters. You cannot specify a generic volume serial. A volume serial number is required and must follow the DELETEVOLUME subcommand.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

DELETEVOLUME Subcommand

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEVRS Subcommand

DELETEVRS: Deleting Vital Record Specifications

Use the DELETEVRS subcommand as shown in Figure 164 to delete a vital record specification. Specify the DSNAME operand to delete a data set vital record specification. To request that a vital record specification that matches both job name and data set name is deleted, specify JOBNAME and DSNAME. Specify the VOLUME operand to delete a volume vital record specification. Specify the NAME operand to delete a NAME vital record specification. When you delete a vital record specification in a chain, DFSMSrmm does not check whether it points to another vital record specification (with the NEXTVRS operand), or whether it is pointed to by another vital record specification.

When a vital record specification is deleted, no data set or volume information is changed. During the next vital records processing run, DFSMSrmm uses only the remaining vital record specifications to apply policies. If the data set or volume matches to another remaining vital record specification, DFSMSrmm applies those policies. If the data set or volume does not match to any vital record specifications, and is no longer retained by a vital record specification, the data sets are eligible for expiration processing.

Example

Task: Delete the volume vital record specification for the volume 8E1U02.

Command:

```
RMM DELETEVRS VOLUME(8E1U02)
```

Syntax

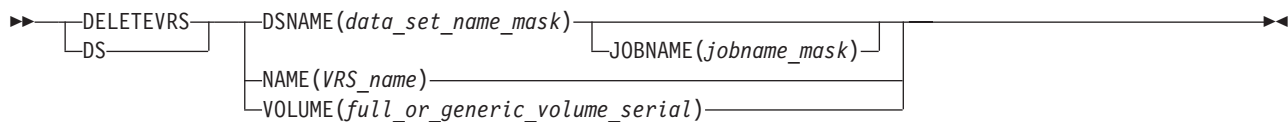


Figure 164. DELETEVRS Syntax Diagram

Operands

DSNAME(data_set_name_mask)

Specifies a data set name mask defined in a vital record specification. You can specify a fully qualified, generic or GDG base data set name. In addition to normal data set naming conventions, you can use any of the masking characters you used to define the data set name vital record specification in the ADDVRS subcommand with the DSNAME operand. See “ADDVRS: Adding a Vital Record Specification” on page 255 for more information.

You can also specify an SMS management class name, a vital record specification management value, or the reserved data set name masks, ABEND or OPEN. An SMS management class name or a vital record specification management value can be eight alphanumeric characters, beginning with an alphabetic character, and must follow standard MVS data set naming conventions. This name must be a single qualifier, and is already assigned by your installation.

For example, specify DSNAME('M99000').

DELETEVRS Subcommand

DSNAME is mutually exclusive with the NAME and VOLUME operands.

JOBNAME(*jobname_mask*)

A job name is one to eight alphanumeric characters or \$, #, or @. Job name must start with an alphabetic character, \$, #, or @. You can specify a specific jobname or a jobname mask. Use % to match any one character and * to match any character string in the mask. This operand is optional. Specify JOBNAME if you want to delete a vital record specification with a jobname mask.

NAME(*VRS_name*)

Specifies the one to eight alphanumeric or national character name of a vital record specification. DFSMSrmm does not check to see if this vital record specification is linked to another vital record specification. NAME is mutually exclusive with the DSNAME and VOLUME operands.

VOLUME(*full_or_generic_volume_serial*)

Specifies the serial number of the volume for which a vital record specification is defined. You can specify a full or a generic volume serial number. A full volume serial number is one to six alphanumeric characters, \$, #, or @, or special characters. A generic volume serial number is one to five alphanumeric characters, \$, #, or @, or special characters followed by an asterisk.

VOLUME is mutually exclusive with the DSNAME and NAME operands.

Example

Task: Delete a data set vital record specification that retained all backup data grouped by the creating job name.

Command:

```
RMM DELETEVRS DSNAME('**.BACKUP') JOBNAME(*)
```

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

GETVOLUME Subcommand

GETVOLUME: Requesting and Assigning Scratch Volumes

Use the GETVOLUME subcommand as shown in Figure 165 on page 327 to request a scratch volume and assign it to an owner defined to DFSMSrmm. The default owner is the owner who issues the command.

You can request volumes from specific pools by specifying a pool ID. If you use an asterisk to specify a pool ID, DFSMSrmm assigns you a volume from the system default scratch pool.

After DFSMSrmm assigns a volume, it sets the volume status to USER, meaning the volume can be overwritten by any data set. DFSMSrmm prevents you from reading any existing data on a scratch volume obtained using the GETVOLUME subcommand.

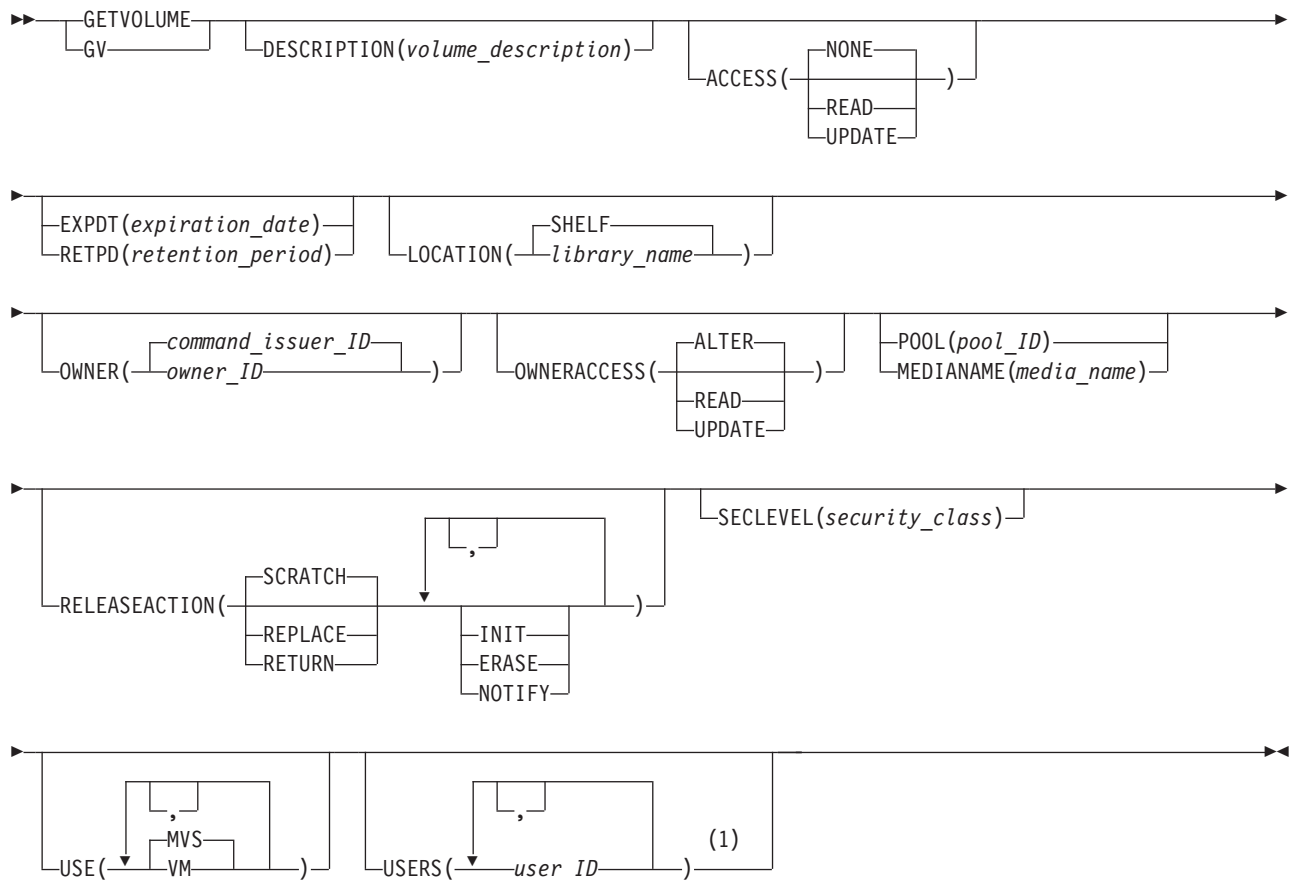
Example

Task: Request a volume from the default scratch pool and set a retention period of 30 days for it.

Command:

```
RMM GETVOLUME RETPD(30)
```

Syntax



Notes:

1. You can specify a maximum of 12 user IDs.

Figure 165. GETVOLUME Syntax Diagram

Operands

ACCESS(NONE,READ,UPDATE)

Specifies user access to a volume. Specify a value to define the access level for users defined as users who can access this volume. You can specify one of the following:

NONE users do not have access to the volume

READ users have only read access to the volume

UPDATE

users have write access to the volume

The default is NONE.

DESCRIPTION(text)

Specifies descriptive text about the volume. Descriptive text is one to thirty characters enclosed in single quotes if it contains any special characters or blanks.

GETVOLUME Subcommand

EXPDT(*expiration_date*)

Specifies date the volume should be considered for release. Specify the year and day in one of two forms:

- *yyddd*, where *yy* is the last two-digit number for the year and *ddd* is the three-digit number for the day of the year
- *yyyy/ddd*, where *yyyy* is the four-digit number for the year and *ddd* is the three-digit number for the day of the year. The slash is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the *yyyy/ddd* format. If you use the *yyddd* format, DFSMSrmm defaults to the 20th century.

EXPDT is mutually exclusive with RETPD.

LOCATION(**SHELF**,*library_name*)

Specifies a library from which the volume should be chosen. Specify one of the following:

SHELF

To indicate that the volume should come from a non-system-managed library. This is the default.

library_name

To indicate that the volume should come from a specific system-managed library. A library name is one to eight alphanumeric characters, \$, #, or @, starting with a non-numeric character, and must be previously defined in the TCDB as a system-managed library.

MEDIANAME(*media_name*)

Specifies the volume's media name. The media name allows you to specify the type or shape of media. Media names are defined by your location and must be one to eight characters.

Use the LISTCONTROL subcommand to display media names for your installation. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information.

The default value is 3480.

OWNER(*owner*)

Specifies the owner ID of the volume's owner. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. We suggest that you use a RACF user ID or RACF group name.

OWNERACCESS(**ALTER,READ,UPDATE**)

Specifies the type of access the owner has to the volume. Depending upon your installation's RACF and DFSMSrmm security options, you can use OWNERACCESS with OWNER to define the initial RACF volume profile access. The value can be READ, UPDATE or ALTER. The default value is ALTER.

POOL(*pool_ID*)

Specifies a pool ID for a group of shelf locations in the removable media library from which DFSMSrmm assigns the volume. The value is one to five characters followed by an asterisk. If you do not specify a pool ID, DFSMSrmm assigns a volume from the default scratch pool.

GETVOLUME Subcommand

Pool IDs are defined by your installation. You can view information about pools by using the LISTCONTROL subcommand with the VLPOOL operand. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333 for more information.

RELEASEACTION(SCRATCH,REPLACE,RETURN,INIT,ERASE, NOTIFY)

Specifies the actions to be taken when the volume is eligible for release. RELEASEACTION can be specified as a list of keywords separated by commas.

You can specify one of the following:

SCRATCH

To request that DFSMSrmm return the volume to scratch status.

REPLACE

To request that the volume be replaced with a new volume and returned to scratch status.

RETURN

To request that the volume be returned to its owner.

SCRATCH, REPLACE and RETURN are mutually exclusive. The default is SCRATCH.

You can specify any or all of the following, separated by commas:

INIT

To request that DFSMSrmm initialize the volume.

ERASE

To request that DFSMSrmm erase the volume.

NOTIFY

To request that DFSMSrmm automatically notify the owner when the volume is released.

For example, you can specify multiple actions as follows:

RELEASEACTION(SCRATCH,INIT,NOTIFY)

RETPD(*retention_period*)

Specifies the number of days DFSMSrmm retains the volume before considering it for release. The value is one to four decimal digits. The retention period is added to today's date to compute the EXPDT.

SECLEVEL(*security_class*)

Specifies the volume's security class. This value is one to eight characters and must be defined by your installation.

Use the LISTCONTROL subcommand with the SECCLS operand to display the security classes defined for your location. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333 for more information.

USE(MVS,VM)

Specifies the operating systems where the volume can be used. You can specify VM, MVS or both. Specify both as MVS,VM with a comma as a separator. The default is MVS.

USERS(*user_ID,user_ID...*)

Specifies the user IDs and group names of users that are allowed to access the

GETVOLUME Subcommand

volume as defined by the ACCESS keyword. You can specify a maximum of twelve user IDs separated by blanks or commas.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTBIN: Displaying Information about a Shelf Location

The LISTBIN subcommand is an alias for the LISTRACK subcommand. See “LISTRACK: Displaying Information about a Shelf Location” on page 349 for the combined description of the LISTRACK and LISTBIN operands.

Use the LISTBIN subcommand as shown in Figure 167 to display information about a single shelf location defined to DFSMSrmm. DFSMSrmm defines shelf space in the storage location as bin numbers.

Use the RMM SEARCHBIN subcommand to request a list of bin numbers defined to DFSMSrmm. See “SEARCHBIN: Creating a List of Bin Numbers” on page 358 for more information.

Example

Task: Request information recorded by DFSMSrmm about the shelf location in the storage location, MYLOC, identified by bin number BIN100, with a media name of SQUARE.

Command:

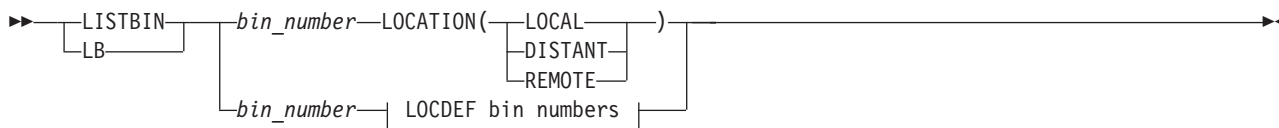
```
RMM LISTBIN BIN100 LOCATION(MYLOC) MEDIANAME(SQUARE)
```

Output: DFSMSrmm displays information as that shown in Figure 166:

```
Rack = BIN100  Volume = VOL000  Status = IN USE  Location = MYLOC
Medianame = SQUARE
```

Figure 166. Sample LISTBIN Output

Syntax



LOCDEF bin numbers:

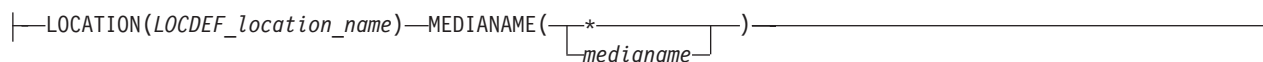


Figure 167. LISTBIN Syntax Diagram

Operands

bin_number

Specifies a shelf location in a storage location. When used with the built-in storage location names, LOCAL, DISTANT, and REMOTE, a bin number is six numeric characters. You must specify leading zeros. When used with installation defined storage locations, a bin number is six alphanumeric or national characters and MEDIANAME must be specified.

A bin number is required and must immediately follow the LISTBIN subcommand.

LISTBIN Subcommand

LOCATION(LOCAL,DISTANT,REMOTE,LOCDEF_location_name)

Specify to request information about a shelf location in a specific storage location. Specify a DFSMSrmm built-in storage location name, DISTANT, LOCAL, or REMOTE.

LOCDEF_location_name. can be any name up to eight characters long. For an installation defined storage location, MEDIANAME must also be specified.

The storage location name does not have to be one that is currently defined using the LOCDEF command. If you do not specify the LOCATION operand, DFSMSrmm lists information about a rack number in the removable media library.

MEDIANAME(medianame,*)

Defines the media which the bin(s) being displayed. MEDIANAME is required when you request a display of an installation defined storage location.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm return and reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTCONTROL: Displaying Parmlib Options and Control Information

Use the LISTCONTROL subcommand as shown in Figure 169 on page 335 to display information in the control record of the control data set and information defined in the EDGRMMxx parmlib member. Figure 168 on page 334 shows the information DFSMSrmm displays when you issue the RMM LISTCONTROL subcommand. If you specify LISTCONTROL without any operands, DFSMSrmm displays only the control record information in the control data set.

To use the LISTCONTROL subcommand, access to the RACF profile STGADMIN.EDG.LISTCONTROL is required when the profile is defined. If the RACF profile STGADMIN.EDG.LISTCONTROL is not defined, READ access to the RACF profile STGADMIN.EDG.MASTER is required. See the *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for information.

Example

Task: Display your installation's options and rules, restricting the information displayed to the control record information and system options only.

Command:

```
RMM LISTCONTROL CNTL OPTION
```

Output: DFSMSrmm displays information such as that shown in Figure 168 on page 334:

LISTCONTROL Subcommand

```
Control record:
Type = MASTER      Create date = 02/24/1997  Create time = 23:08:42
Journal utilisation = 15% (75% threshold)
Last backup:                               Last expiration processing:
  Date = 07/07/1997  Time = 02:20:16        Date = 07/14/1997  Time   = 00:03:16
Last report extract:                         Last store update:
  Date = 07/14/98    Time = 00:03:16        Date = 07/14/1997  Time   = 00:03:16
Last scratch procedure:                     Last VRS processing:
  Date = 04/08/1997  Time = 23:19:08        Date = 07/14/1997  Time   = 00:03:16
Last Catalog synchronize
  Date = 04/08/1997  Time   = 23:19:08

Rack numbers      = 32897                    Empty racks      = 11745
LOCAL store bins  = 38                      Empty LOCAL bins = 30
DISTANT store bins = 764                    Empty DISTANT bins = 754
REMOTE store bins = 35                      Empty REMOTE bins = 29

Control functions in progress:
Backup           = N  Restore           = N
Verify          = N  Expiration         = N
Report Extract  = N  Disaster Store     = N
VRS             = N  Synchronize        = N

System options:
Parmlib suffix   = UK
Operating mode   = P    Retention period: Default = 0      Maximum = 9999
                                   Catalog = 12      hours

Control data set name      = D088562.TT.MASTER
Journal file data set name =
Catalog SYSID             = *
Scratch procedure name     = S1ATL008
Backup procedure name      = SUB
IPL date check = N        Date format   = A        RACF support   = P
SMF audit      = 248      SMF security  = 249      CDS id           = EXPR
MAXHOLD value  = 100      Lines per page = 54      System ID        = W98M
BLP            = NORMM    TLCS V1       = N        Notify          = N
Uncatalog      = N        VRS job name   = 2        Message case     = M
MASTER overwrite= LAST    Accounting    = J        VRS selection    = NEW
VRS change     = INFO     VRSMIN action  = FAIL     VRSMIN count     = 2147
Disp DD name    = RMMDISP  Disp msg ID   = EDG4054I
```

Figure 168. Sample LISTCONTROL Output

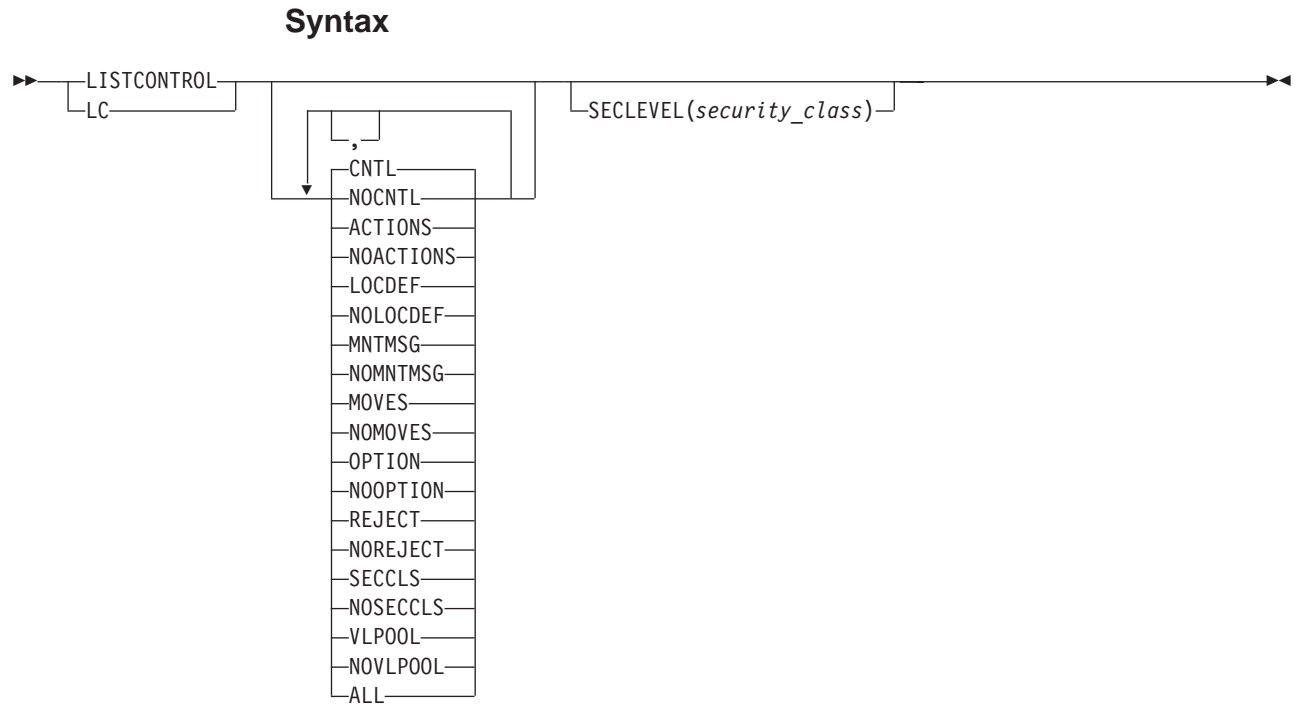


Figure 169. LISTCONTROL Syntax Diagram

Operands

ACTIONS

Specify to display information about outstanding volume actions. DFSMSrmm uses the following status values:

Pending

An action is outstanding and should be taken.

Confirmed

An action has been confirmed as having been taken.

Complete

Inventory management has been run and all volumes updated to show that the action has been taken.

Unknown

The status of an action cannot be determined.

ALL

Use to display all options and rules defined for your installation. Specifying ALL is the same as specifying the operands ACTIONS, CNTL, LOCDEF, MNTMSG, MOVES, OPTION, REJECT, SECCLS, and VLPOOL. You can use any of the following operands with ALL to limit the amount of information displayed: NOCNTL, NOACTIONS, NOLOCDEF, NOMNTMSG, NOMOVES, NOOPTION, NOREJECT, NOSECCLS, NOVLPOOL.

CNTL

Use to display the control record information in the DFSMSrmm control data set, describing the inventory for the removable media library. This information includes inventory management and backup dates, and inventories for rack and bin numbers.

CNTL is the default.

LISTCONTROL Subcommand

LOCDEF

Use to display all locations defined using the LOCDEF command in the EDGRMMxx parmlib member. Default values for built-in storage location names such as LOCAL, DISTANT, REMOTE, and location SHELF are also displayed. Values for specific system-managed libraries are displayed if they are defined in LOCDEF parameters.

MOVES

Use to display information about outstanding volume moves. DFSMSrmm uses the following status values:

Pending

a move is outstanding and should be completed.

Confirmed

a move has been confirmed as having been completed.

Complete

DFSMSrmm inventory management has been run and all volumes have been updated to show that the move has been completed.

Unknown

the status of a move cannot be determined.

MNTMSG

Use to display the mount/fetch message definitions that are currently in use.

NOACTIONS

Use to prevent outstanding volume actions from being listed when you specify LISTCONTROL ALL.

NOCNTL

Use to prevent the control record information from being displayed when you specify LISTCONTROL ALL.

NOLOCDEF

Use to prevent the location definition information from being displayed when you specify LISTCONTROL ALL.

NOMOVES

Use to prevent outstanding volume moves from being listed when you specify LISTCONTROL ALL.

NOMNTMSG

Use to prevent the mount/fetch message definitions from being displayed when you specify LISTCONTROL ALL.

NOOPTION

Use to prevent system options from being displayed when you specify LISTCONTROL ALL.

NOREJECT

Use to prevent volumes that are unavailable on the system from being displayed when you specify LISTCONTROL ALL.

NOSECCLS

Use to prevent security classes from being displayed when you specify LISTCONTROL ALL.

NOVLPOOL

Use to prevent volume pool details from being displayed when you specify LISTCONTROL ALL.

LISTCONTROL Subcommand

OPTION

Use to display the system options that are currently in use.

REJECT

Use to display rack number and access information about volumes that are unavailable on the system.

SECCLS

Use to display the security classes defined for your installation.

SECLEVEL(*security_class*)

Use to display information about security classes. Specify a security class ID. The value must be one to eight characters and must correspond to one of the security classes defined for your installation.

VLPOOL

Use to display information about pools defined for your installation.

Example

Task: Display your installation's options and rules excluding control record information and start up options.

Command:

```
RMM LISTCONTROL ALL NOCNTL NOOPTION NOLOCDEF
```

Output: DFSMSrmm displays information such as that shown in Figure 170 on page 338.

LISTCONTROL Subcommand

Security classes:

Number	Name	SMF	MSG	Erase	Description
1	CLASS01	N	N	N	FOR SECURITY CLASSIF. TESTING
2	CLASS02	Y	N	N	FOR SECURITY CLASSIF. TESTING
3	CLASS03	N	N	Y	FOR SECURITY CLASSIF. TESTING
4	CLASS04	Y	N	Y	FOR SECURITY CLASSIF. TESTING
5	CLASS05	N	Y	N	FOR SECURITY CLASSIF. TESTING
9	UNCLASS	N	N	N	UNCLASSIFIED
10	IUO	N	N	N	XXX INTERNAL USE ONLY
11	IC	N	N	N	XXX CONFIDENTIAL
12	ICR	Y	Y	Y	XXX CONFIDENTIAL RESTRICTED

Volume Pools:

Pool	System	RA CF	Ty	Expdt pe	Pool name	Media name	Description
*	BTLS110	N	S	0	SCRCH2	3490	BTLS POOL
P4L33*		N	S	0		TAPE	DIALOGUE VOLUME TESTS
P4L41*		N	S	0		TAPE	DIALOGUE VOLUME TESTS
P4L42*		N	R	0		TAPE	DIALOGUE VOLUME TESTS
P4L43*		N	S	0		TAPE	DIALOGUE VOLUME TESTS
R2*		N	R	0		TAPE	DIALOGUE RACK TESTS
R5*		N	R	0		TAPE	DIALOGUE PRODUCT TESTS
S3*		N	S	0		TAPE	DIALOGUE DATASET TESTS
TE520*		Y	S	0		3420	TEST T14000 3420S
T1*		N	R	0		3480	DIALOGUE OWNER TESTS
T3*		N	R	0		TAPE	DIALOGUE DATASET TESTS
T4*		N	R	0		TAPE	DIALOGUE VOLUME TESTS
11501*		Y	S	0		3480	TEST T14000 SPECIFIC
1150*	DGH	Y	R	0		3480	SYSTEM TEST SCRATCH VOLS
9990*	DGH	Y	S	N		3480	TEST T12000 SPECIFIC
*		N	S	0		3480	DEFAULT VOLUME POOL

Figure 170. Sample LISTCONTROL Output (Part 1 of 2)

LISTCONTROL Subcommand

```
Mount/Fetch messages:
Message ID    ID  Volume Rack
-----
IEC501A      1   15    999
IEC502E K    1   15    999
IEF233A      1   15     15
IEF233D      1   15    998
IEF234E D    1   15     15
```

NO INSTALLATION DEFINED REJECT PREFIXES

```
Action  Status
-----
ERASE    Pending
INIT     Pending
NOTIFY   Pending
SCRATCH  Pending
REPLACE  Unknown
RETURN   Unknown
```

```
Source  Target  Status
-----
LIBRARY REMOTE  Unknown
REMOTE  LIBRARY Unknown
LIBRARY LOCAL  Unknown
LOCAL   LIBRARY Unknown
LOCAL   DISTANT Unknown
LIBRARY DISTANT Unknown
DISTANT LIBRARY Unknown
```

Figure 170. Sample LISTCONTROL Output (Part 2 of 2)

Task: List the data set name masks for a security level defined as IC.

Command:

```
RMM LISTCONTROL SECLEVEL(IC)
```

Output: DFSMSrmm displays information such as shown in Figure 171:

```
Security rules for level 11
Name = IC          SMF = N          MSG = N
Erase = N  Description = XXX CONFIDENTIAL
Data set name Mask(s)
-----
SYS1.IC.IC.**
```

Figure 171. Sample LISTCONTROL Output

LISTCONTROL Subcommand

Task: List the location definitions defined for your installation.

Command:

```
RMM LISTCONTROL LOCDEF
```

Output:

If no location definitions are defined in EDGRMMxx parmlib, then issuing LISTCONTROL LOCDEF produces the output shown in Figure 172:

Location	Def	Mgtype	Ltype	Priority	Medianames
	N		AUTO	4800	
	N		MANUAL	4900	
DISTANT	N		STORE	200	
LOCAL	N		STORE	300	
REMOTE	N		STORE	100	
SHELF	N			5000	

Figure 172. LISTCONTROL LOCDEF Output - No LOCDEFs Defined

Table 30 describes the columns in the LISTCONTROL subcommand output.

Table 30. LISTCONTROL Subcommand Output

Column	Description
Location	The location name, in ascending alphanumerical order.
Def	Shows Y or N for whether a LOCDEF command was found for this location.
Mgtype	Shows any MANAGEMENTTYPE setting for the location. Mgtype can be: BINS, NOBINS, or blank. BINS indicates the location is shelf-managed. NOBINS indicates the location is not shelf-managed.
Ltype	Is the location type. AUTO, MANUAL, STORE, or blank for location SHELF.
Priority	Is the current relative priority of the location.
Medianames	Provides all the media names specified for the location.

Figure 173 shows information that DFSMSrmm displays when there are location definitions in place.

Location	Def	Mgtype	Ltype	Priority	Medianames
	N		AUTO	4800	
	N		MANUAL	4900	
COALMINE	Y	BINS	STORE	50	3480,*
DISTANT	Y	BINS	STORE	200	3490,3420,*,3480,348X, SQUARE
LIBDEF3	Y		AUTO	4000	
LIBDEF4	Y		AUTO	4200	
LOCAL	N		STORE	300	
OLDMINE	Y	NOBINS	STORE	2100	3420
REMOTE	Y	NOBINS	STORE	100	*
SHELF	Y			6000	

Figure 173. LISTCONTROL LOCDEF Output - LOCDEFs Defined

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTDATASET Subcommand

LISTDATASET: Displaying Information about a Data Set

Use the LISTDATASET subcommand as shown in Figure 175 on page 343 to display information about a single data set defined to DFSMSrmm. You must specify the data set name and the volume serial number where the data set resides. If the data set is not the first data set on the volume, you must also specify a sequence number.

The output from the LISTDATASET subcommand includes:

- Creating job name
- Data set retention date
- Storage group
- Storage class
- Data class
- Matching VRS type:
 - DATASET type indicates the data set is retained by a DSNAME type vital record specification.
 - SMSMC type indicates the data set is retained by a vital record specification matching its SMS management class.
 - VRSMV type indicates the data set is retained by a vital record specification matching its VRS management value.
 - DSN/MV type indicates the data set is retained by a DSNAME type vital record specification and a management value VRS defined with WHILECATALOG.
- Matching VRS name
- Vital record status

Use the SEARCHDATASET subcommand to list all data sets on a volume. See “SEARCHDATASET: Creating a List of Data Sets” on page 362 for more information.

Example

Task: Display information recorded by DFSMSrmm for a data set named NISHINAL.TEST.ORDER that resides on volume BC0047 and is the first data set on the volume.

Command:

```
RMM LISTDATASET 'NISHINAL.TEST.ORDER' VOLUME(BC0047) SEQ(1)
```

If NISHINAL is your own TSO PROFILE PREFIX, you can also enter:

```
RMM LISTDATASET TEST.ORDER VOLUME(BC0047)
```

Output: DFSMSrmm displays output such as that shown in Figure 174 on page 343:

LISTDATASET Subcommand

```
Data set name = NISHINAL.TEST.ORDER
Job name      = PERS01
Volume        = BC0047          Physical file sequence number = 1
Device number = 05FB            Owner          = #281753 Data set sequence = 1
Create date   = 04/01/1996 Create time = 10:12:45 System ID      = BCRVSC1
Block size    = 32760          Block count = 900
Logical Record Length = 0      Record Format = U
Date last written = 04/01/1996 Date last read = 04/01/1996
Step name      = RDS           DD name        = TAPE
Management class =             VRS management value =
Storage group   =             VRS retention date =
Storage class   =             VRS retained      = NO
Data class      =             ABEND while open  = NO
                                   Catalog status = UNKNOWN

Primary VRS details:
  Name          =
  Job name      =          Type                  =
  Subchain NAME =          Subchain start date   =

Secondary VRS details:
  Value or class =
  Job name      =
  Subchain NAME =          Subchain start date   =
Security Class  = IUO          Description      = IBM INTERNAL USE ONLY
```

Figure 174. Sample LISTDATASET Output

Syntax

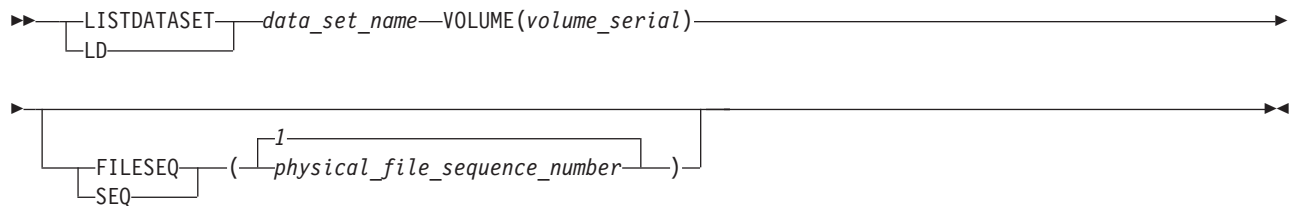


Figure 175. LISTDATASET Syntax Diagram

Operands

data_set_name

Specifies the name of the data set about which you want to view information. The name follows standard MVS naming conventions for data sets, must not include a member name, and cannot be greater than eight characters. This operand is required and must immediately follow the LISTDATASET subcommand.

FILESEQ/SEQ(1,physical_file_sequence_number)

Specifies the relative position of the data set on the volume. The minimum allowable decimal value is 1. The maximum allowable decimal value is 9999.

The default value is 1.

VOLUME(volume_serial)

Specifies the serial number of the volume where the data set resides. A volume serial number is one to six alphanumeric characters, \$, #, or @, or special characters. This operand is required.

LISTDATASET Subcommand

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTOWNER: Displaying Information about an Owner

Use the LISTOWNER subcommand as shown in Figure 177 to display information about a single owner defined to DFSMSrmm.

Example

Task: Request information recorded by DFSMSrmm about the owner whose owner ID is GILLESPB.

Command:

```
RMM LISTOWNER GILLESPB
```

Output: DFSMSrmm displays information such as that shown in Figure 176:

```

Owner          = GILLESPB
Last name      = Surname           First names = First name
Department     = Department
Address        = Address Line One
                Address Line Two
                Address Line Three
Telephone:
Internal       = 123-5678           External    = 987654-5678
Electronic mail:
Userid         = USERID           Node         = NODE

Volumes        = 18

```

Figure 176. Sample LISTOWNER Output

Syntax

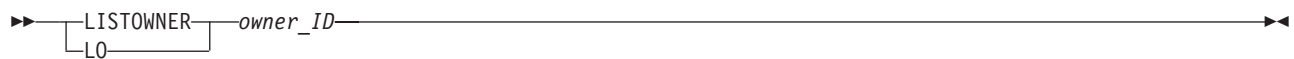


Figure 177. LISTOWNER Syntax Diagram

Operands

owner_ID

Specifies the owner ID of the owner for whom you are requesting information. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. This operand is required.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

LISTOWNER Subcommand

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTPRODUCT: Displaying Information about a Software Product

Use the LISTPRODUCT subcommand as shown in Figure 179 to display information about a software product defined to DFSMSrmm. You must specify the software product number and, optionally, its version. If you do not specify the version, the default is V01R01M00, Version 1, Release 1, Modification Level 0.

Use the SEARCHPRODUCT subcommand to create lists of software products defined to DFSMSrmm. See “SEARCHPRODUCT: Creating a List of Software Products” on page 369 for more information.

Example

Task: Request information recorded by DFSMSrmm about a software product with the product number PROD01, version 1.1.0.

Command:

```
RMM LISTPRODUCT PROD01
```

Output: DFSMSrmm displays information such as that shown in Figure 178:

```
Product Number = PROD01    Level = V01R01M00    Owner = KUEHNW
Name           = Product One
Description    =

Volume      Rack      Feature Code
-----
VOL100     RAC200      1234
VOL101     RAC201      2234
VOL102     RAC202      3234
VOL103     RAC203      4234
VOL104     RAC204      5234
```

Figure 178. Sample LISTPRODUCT Output

Syntax



Figure 179. LISTPRODUCT Syntax Diagram

Operands

LEVEL(*software_product_version*)

Specifies the software product's version. Specify the version in the form, VnnRnnMnn, indicating the version, release and modification level. 'nn' is two numbers in the range 00 to 99.

The default value is V01R01M00.

software_product number

Specifies the number of the software product. A software product number is one to eight characters enclosed in single quotes if it contains any special characters or blanks. This operand is required and must immediately follow the LISTPRODUCT subcommand.

LISTPRODUCT Subcommand

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTRACK: Displaying Information about a Shelf Location

This section describes the combined description for the LISTRACK subcommand and its alias LISTBIN. See “LISTBIN: Displaying Information about a Shelf Location” on page 331 for information about using the RMM LISTRACK subcommand alias.

Use the LISTRACK subcommand as shown in Figure 181 to display information about a single shelf location defined to DFSMSrmm. DFSMSrmm defines shelf space in the removable media library as rack numbers and bin numbers in a storage location. When you request information about a shelf location, you must specify the full six character rack or bin number. Specify a location when you request information about a bin number in a built-in or installation defined storage location.

Use the SEARCHRACK subcommand or SEARCHBIN subcommand to request lists of rack or bin numbers defined to DFSMSrmm. See “SEARCHRACK: Creating a List of Shelf Locations” on page 372 and “SEARCHBIN: Creating a List of Bin Numbers” on page 358 for more information.

Example

Task: Request information recorded by DFSMSrmm about the shelf location in the removable media library identified by rack number RAC100.

Command:

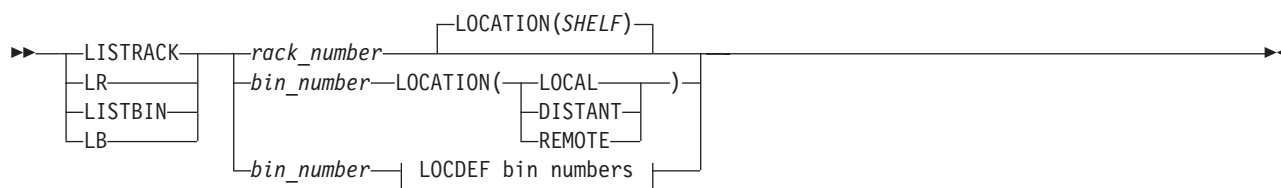
```
RMM LISTRACK RAC100
```

Output: DFSMSrmm displays information such as that shown in Figure 180: :

```
RMM LISTRACK RAC100
Rack = RAC100 Volume = VOL000 Status = IN USE Location = SHELF
Medianame = 3480
```

Figure 180. Sample LISTRACK Output

Syntax



LOCDEF bin numbers:

```
|LOCATION(LOCDEF_location_name)—MEDIANAME(*medianame)|
```

Figure 181. LISTRACK Syntax Diagram

LISTRACK Subcommand

Operands

bin_number

Specifies a shelf location in a storage location. A bin number is six numbers for built-in storage locations and six alphanumeric or national characters for an installation defined storage location. You must specify leading zeros.

A rack or bin number is required and must immediately follow the LISTTRACK or LISTBIN subcommand.

LOCATION(SHELF,LOCAL,DISTANT,REMOTE,LOCDEF_location_name)

Specifies the location you want to list. Use SHELF to list shelf locations in your removable media library. The DFSMSrmm built-in storage location names are: LOCAL, DISTANT, and REMOTE. *LOCDEF_location_name* can be a name up to eight characters long.

You must use MEDIANAME with installation defined storage location names.

MEDIANAME(*medianame*,*)

Defines the media name of the rack number or bin number to be listed. *medianame* can be any name up to eight characters.

rack_number

Specifies a shelf location in the removable media library. A rack number is six alphanumeric or national characters. You cannot use a generic rack number.

A rack or bin number is required and must immediately follow the LISTTRACK or LISTBIN subcommand.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTVOLUME: Displaying Information about a Volume

Use the LISTVOLUME subcommand as shown in Figure 183 on page 352 to display information recorded by DFSMSrmm for a single volume. You must specify a volume serial number. You can optionally limit the amount of information DFSMSrmm displays.

DFSMSrmm displays access information, volume serial information, volume statistics, and storage information for the volume. For example, DFSMSrmm displays:

- Owner and user authorization
- Identifier, such as a user ID, of the last owner, product, or utility to change information about the volume
- Access list
- Operating system usage
- The number of data sets on a volume
- Volume chaining information
- Software products residing on the volume
- The volume's current location
- Movement information
- Storage location data and bin numbers
- Assigned date
- Status
- Release actions
- Security classification
- Expiration date
- The name of the first data set on the volume
- The next location to which the volume will move
- The bin numbers associated with the volume
- Retention date
- Data set recording

Use the SEARCHVOLUME subcommand to create lists of volumes defined to DFSMSrmm. See “SEARCHVOLUME: Creating a List of Volumes” on page 376 for more information.

Example

Task: Request the volume serial information for the volume with serial number 003186.

Command:

```
RMM LISTVOLUME 003186 VOL ACCESS
```

Output: DFSMSrmm displays information such as that shown in Figure 182 on page 352:

LISTVOLUME Subcommand

```
Volume Information:
Volume = 003186      Rack   = 003186   Owner = RMMUSER   Jobname =
Type = PHYSICAL
Create Date   = 1997/029   Create Time   = 05:15:32
Assigned Date = 1997/029   Assigned Time = 05:15:33
Expiration Date = 1997/029   Original     =
Retention Date =
Data set name =
Volume Status:
Status = MASTER      Availability =
Current Label version = 3      Required label version = 4
Media Information:
Density = *      Type = CST      Format = 18TRACK   Compaction = YES
Special attributes = RDCOMPAT
Action on Release:
Scratch immediate = N   Expiry date ignore = N
Scratch = Y   Replace = N   Return = N   Init = N   Erase = N   Notify = N
Actions pending:
Scratch = N   Replace = N   Return = N   Init = N   Erase = N   Notify = N
Storage group =
Loan location =          Account =
Description   =
Security class =          Description =
```

Figure 182. Sample LISTVOLUME Output

Syntax

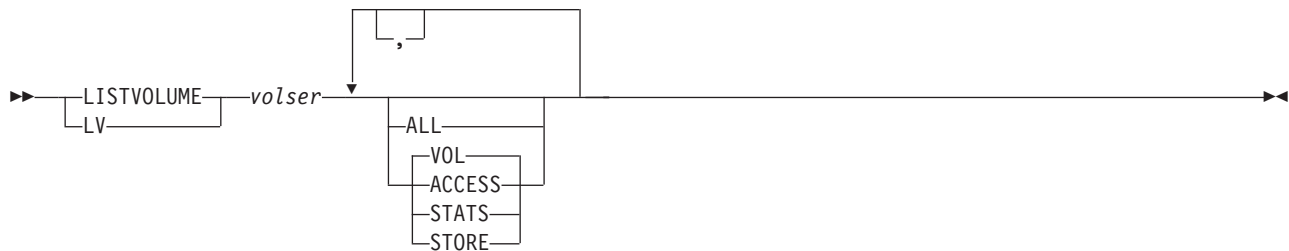


Figure 183. LISTVOLUME Syntax Diagram

Operands

ACCESS

Specify to request the access information for the volume. Access information includes owner and user authorization, last change information, access list, and operating system usage.

ALL

Specify to request all the information recorded by DFSMSrmm for the volume. Specifying ALL is equivalent to specifying the operands VOL, ACCESS, STATS and STORE.

STATS

Specify to request the volume statistics. Volume statistics include the number of data sets on a volume, volume chaining information, and products that reside on the volume.

STORE

Specify to request the storage information for the volume. Storage location information includes the current location of a volume, movement information, store data and bin numbers.

VOL

Specify to request volume serial information and status. This includes assigned date, status, release actions, security classification, expiration date, and the name of the first data set.

VOL is the default.

volser

Specifies the volume's serial number. A fully qualified serial number is one to six alphanumeric characters, \$, #, or @, or special characters. You cannot specify a generic volume serial. A volume serial number is required and must follow the LISTVOLUME subcommand.

Example

Task: Request the volume statistics and information about storage for the volume with serial number DSP000.

Command:

```
RMM LISTVOLUME DSP000 STATS STORE
```

Output: DFSMSrmm displays information such as that shown in Figure 184:

```
Statistics:
Number of data sets = 1      Data set recording= ON
Volume Usage(Kb)= 0        Use Count      = 0
Date last read  =          Date last written =
Drive last used = 0BD9
Volume sequence = 1         Media name      = 3480
Previous volume =           Next volume     =
Product Number  =          Level          = V  R  M
Feature code    =
Error counts:
Temporary read = 0          Temporary write = 0
Permanent read = 0          Permanent write = 0

Store Information:
Location      = COALMINE   Destination   =          Intransit    = N
Location type = STORE      Home location = SHELF   Old location = REMOTE
                               Required location = SHELF
Movement tracking date = 22/03/1994      Move mode    = MANUAL
Bin number    = CLM025     Media name    = 3480
Old bin number = 000004    Media name    = *
In container  = PV0021
```

Figure 184. Sample LISTVOLUME Output

Note: The Volume Usage value in the volume statistics information is calculated from the DCBBLKSIZE multiplied by the number of blocks. For data sets created by DFSMSdss, Volume Usage is 0.

LISTVOLUME Subcommand

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTVRS: Displaying Information about a Vital Record Specification

Use the LISTVRS subcommand as shown in Figure 186 on page 356 to display details about a single vital record specification. Specify a data set name when requesting information about a data set vital record specification. You can optionally use the JOBNAME operand when requesting information about a data set vital record specification. Specify a volume serial number when requesting information about a volume vital record specification. Specify a vital record specification name when requesting information about a vital record specification used to link to other vital record specifications.

The values in the Type field in the LISTVRS output are DSN, GDG, or PGDG. These values indicate if the vital record specification was specified with a NOGDG, GDG, or a pseudo-GDG data set name.

Use the SEARCHVRS subcommand to create lists of vital record specifications. See “SEARCHVRS: Creating a List of Vital Record Specifications” on page 392 for more information.

Example

Task: Request information about the data set vital record specification defined for the data set named DATA.SET.ONE.

Command:

```
RMM LISTVRS DSNAME('DATA.SET.ONE')
```

or, if DATA is your own TSO PROFILE PREFIX, you can enter:

```
RMM LISTVRS DSNAME(SET.ONE)
```

Output: DFSMSrmm displays information such as that shown in Figure 185:

```
Data set Mask = DATA.SET.ONE                                Type = DSNAME
Job name Mask = *                                           Retain until expired = NO
Count          = 99999 CYCLES                               Retain while cataloged = NO
Delay          = 0      Days in the HOME location
Store number   = 99999 CYCLES in the HOME location
Priority       = 500

                                     Release Options:
                                     Expiry date ignore = NO
                                     Scratch immediate = YES

Next VRS in chain   = AADN!      using ANDVRS

VRS Owner          = OWN000
Description        = User data sets
Vital Record Specification to be deleted on 31/12/1999
```

Figure 185. Sample LISTVRS Output

LISTVRS Subcommand

Syntax

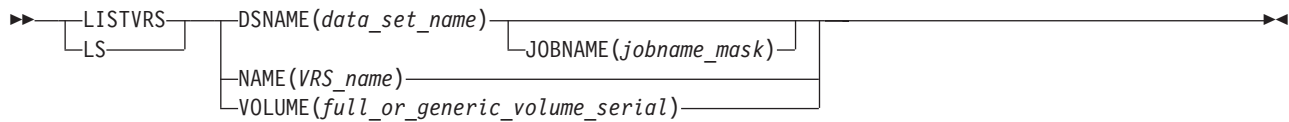


Figure 186. LISTVRS Syntax Diagram

Operands

DSNAME(data_set_name)

Specifies the name of the data set for which the vital record specification is defined. The name follows standard MVS naming conventions for data sets and must not include a member name.

You can also specify an SMS management class name, a vital record specification management value, or the reserved data set name masks, ABEND or OPEN. The name can be eight alphanumeric characters, beginning with an alphabetic character, and must follow standard MVS data set naming conventions. This name must be a single qualifier, and is already assigned by your installation. For example, you can specify DSNAME('M99000').

To list a vital record specification based on job name, you must code both DSNAME and JOBNAME and both must match the corresponding values in the vital record specification. For example, if you want to list the two vital record specifications shown in Figure 187:

If you issue the command shown in Figure 188:

```
RMM ADDVRS DSNAME('A.B') WHILECATALOG
RMM ADDVRS DSNAME('A.B') JOBNAME(BENSJOY) WHILECATALOG
```

Figure 187. Listing Vital Record Specification Information Examples

DFSMSrmm does not list the second vital record specification. You must also

```
RMM LISTVRS DSNAME('A.B')
```

Figure 188. Listing Vital Record Specification Information

specify the JOBNAME operand on the LISTVRS subcommand.

DSNAME is mutually exclusive with the NAME and VOLUME operands.

NAME(VRS_name)

Specifies the eight-character name of a vital record specification. NAME is mutually exclusive with the DSNAME and VOLUME operands.

JOBNAME(jobname_mask)

Specifies the job name for the vital record specification. A job name is one to eight alphanumeric characters or \$, #, or @. The job name must start with an alphabetic character, \$, #, or @. You can specify a specific jobname or a jobname mask. Use % to match any one character and * to match any character string in the mask. The job name mask you specify must exactly match the job name mask in the vital record specification. This operand is optional. Specify JOBNAME to display a vital record specification defined with a jobname mask.

VOLUME(*full_or_generic_volume_serial*)

Specifies the serial number of the volume for which the VRS is defined. A full volume serial number is one to six alphanumeric characters, \$, #, or @, or special characters. You can specify a generic volume serial number if it exactly matches an existing vital record specification.

VOLUME is mutually exclusive with the DSNAME and NAME operands.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHBIN Subcommand

SEARCHBIN: Creating a List of Bin Numbers

Use the SEARCHBIN subcommand as shown in Figure 190 on page 359 to create a list of shelf locations defined to DFSMSrmm. DFSMSrmm defines shelf space in storage locations as bin numbers.

You can restrict how many bin numbers DFSMSrmm lists by specifying the LIMIT operand. DFSMSrmm searches until your limit is reached or until it lists all shelf locations matching your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten shelf locations.

Table 31 shows the information DFSMSrmm returns for each bin number in the list, in the order it is displayed:

Table 31. Information Returned by SEARCHBIN

Table Field Name	Description
Rack	Bin number
Medianame	Type of volume
Volume	Volume serial number
Status	Status of the shelf location (one of EMPTY, INUSE, or SCRATCH)
Location	Location where the volume resides

Example

Task: Create a list of empty bin numbers that are available for use in the DISTANT storage location.

Command:

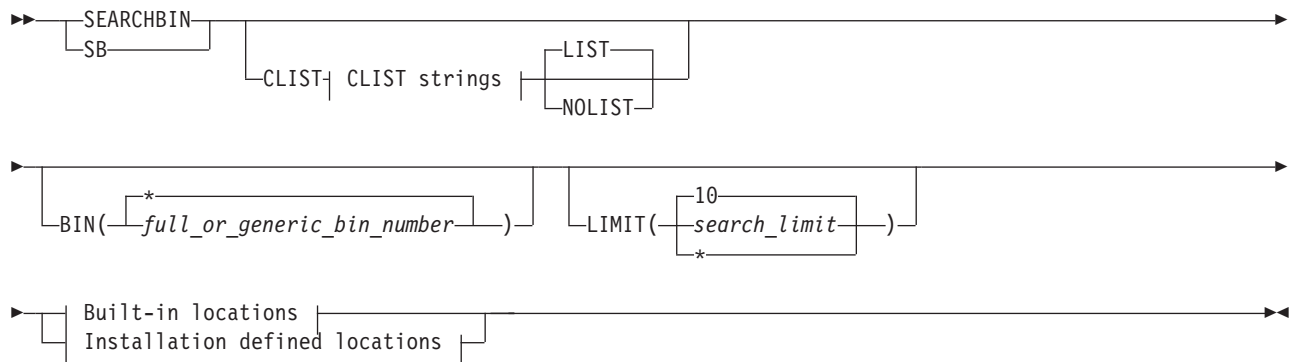
```
RMM SEARCHBIN BIN(A*) LOCATION(DPBINS) MEDIANAME(3480) EMPTY
```

Output: DFSMSrmm displays a list such as the one shown in Figure 189:

Rack	Medianame	Volume	Status	Location
-----	-----	-----	-----	-----
A00001	3480		EMPTY	DPBINS
A00002	3480		EMPTY	DPBINS
A00003	3480		EMPTY	DPBINS
A00004	3480		EMPTY	DPBINS
A00005	3480		EMPTY	DPBINS
A00006	3480		EMPTY	DPBINS
A00007	3480		EMPTY	DPBINS
A00008	3480		EMPTY	DPBINS
A00009	3480		EMPTY	DPBINS
A00010	3480		EMPTY	DPBINS
EDG3203I SEARCH COMPLETE - MORE ENTRIES MAY EXIST				
EDG3012I 10 ENTRIES LISTED				

Figure 189. Sample SEARCHBIN Output

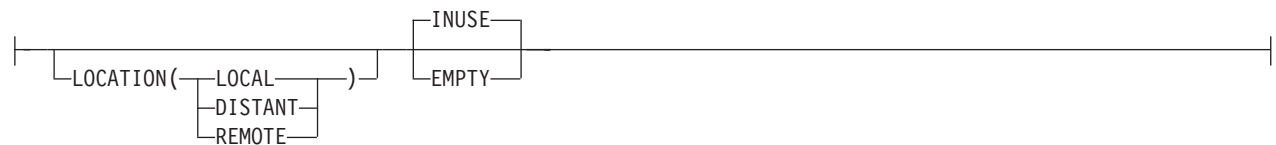
Syntax



CLIST strings:

—(prefix_string,suffix_string)—

Built-in locations:



Installation defined locations:

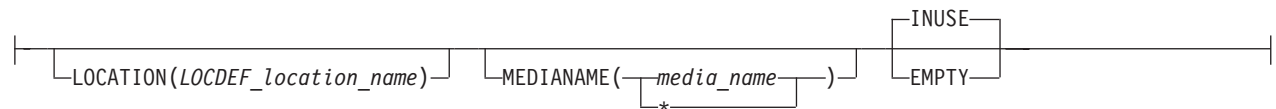


Figure 190. SEARCHBIN Syntax Diagram

Operands

BIN(full_or_generic_bin_number,*)

Specify a full or generic bin_number to define a shelf location in a storage location. A bin number in a built-in storage location is six numeric characters. A bin number in an installation defined storage location is six alphanumeric or national characters. You must also give a location name and media name.

An asterisk tells DFSMSrmm to search through all bin numbers.

CLIST(prefix_string,suffix_string)

Specify CLIST to create a data set of executable commands.

For each shelf location that matches the search criteria, DFSMSrmm writes one or more records in a sequential data set. If RMMCLIST DD is pre-allocated, DFSMSrmm uses this data set for the CLIST output. If RMMCLIST is not pre-allocated, DFSMSrmm uses the data set called 'prefix.EXEC.RMM.CLIST', where *prefix* is a TSO user PROFILE PREFIX value. If you do not change this value, it will be the same as your TSO user ID. If you have specified PROFILE NOPREFIX, DFSMSrmm gets the RACF user ID from the ACEE and uses it as the prefix as long as the RACF user ID is valid; otherwise, DFSMSrmm uses

SEARCHBIN Subcommand

the job name. The data set uses blocked, variable-length records, with a logical record length of 255. If the length of the output record exceeds 255 characters, it is split into multiple records and DFSMSrmm adds a continuation character, +, to all but the last record. If the CLIST data set already exists, DFSMSrmm attempts to use it. If it is a partitioned data set, DFSMSrmm creates or reuses the member TEMPNAME.

Specify *prefix_string* and *suffix_string* to add subcommands and operands to the records in the file. DFSMSrmm returns the data as follows:

```
'prefix_string'bin_number'suffix_string'
```

prefix_string and *suffix_string* can be any characters. You specify CLIST values in prefix and suffix text strings. The total length of each text string cannot exceed 255 characters. Enclose your text string in quotes if the string contains any separator characters, such as blanks and commas. Use a blank or a comma between the text strings when you specify both a prefix and a suffix value. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the *prefix_string* with the data DFSMSrmm returns and the *suffix_string*.

If you do not specify a CLIST value, the file '*prefix*.EXEC.RMM.CLIST', contains only the number of the shelf location for each record.

You can edit the CLIST to remove any shelf locations you do not want in the list, then you can run it at your convenience.

EMPTY

Specify to list empty bin numbers in a storage location. An empty bin number does not contain a volume and is available for use.

INUSE

Specify to list bin numbers in a storage location that are in use.

INUSE is the default.

LIMIT(*search_limit*,*)

Specifies how many entries DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify an asterisk to request a list of all entries matching your search criteria.

The default value is 10.

LIST

Specifies that DFSMSrmm produce a list when the CLIST operand is used.

LIST is mutually exclusive with the NOLIST operand. LIST is the default.

LOCATION(**LOCAL**,**DISTANT**,**REMOTE**,*LOCDEF_location_name*)

Specify to search in a specific storage location. Specify a built-in storage location name, LOCAL, DISTANT, or REMOTE or *LOCDEF_location_name* which is an up to eight character name. For an installation defined storage location, MEDIANAME can also be specified.

The storage location name does not have to be one that is currently defined using the LOCDEF command.

MEDIANAME(*media_name*,*)

Specify to limit the list to shelf locations containing volumes belonging to the same media name. The media name allows you to specify the type or shape of

SEARCHBIN Subcommand

media. They are defined by your installation and must be one to eight characters. You can also use the media name * which is a media name defined in a LOCDEF command.

If you do not specify MEDIANAME, all the bin numbers in the specified location are listed.

Use the LISTCONTROL subcommand to display media names defined for your location. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333 for more information.

NOLIST

Specifies that DFSMSrmm not produce a list when the CLIST operand is used. DFSMSrmm produces only the CLIST output file.

NOLIST is mutually exclusive with the LIST operand. LIST is the default.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHDATASET Subcommand

SEARCHDATASET: Creating a List of Data Sets

Use the SEARCHDATASET subcommand as shown in Figure 192 on page 363 to create a list of data sets that match criteria you specify. You can specify a generic name using the full set of filter masks. Specify a fully-qualified name to restrict the search to data sets that match the name exactly. Specify the FILESEQ operand to restrict the search to data sets at a relative position on the volume. You can also use the JOBNAME operand to restrict the search to data sets created by a particular job name.

You can restrict how many data sets DFSMSrmm lists by specifying the LIMIT operand. DFSMSrmm searches until it reaches your limit or until it finds all data sets matching your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten data sets.

Table 32 shows the information DFSMSrmm returns for each data set in the order it is displayed:

Table 32. Information Returned by SEARCHDATASET

Table Field Name	Description
Data set name	Data set name
Volume	Volume serial number of the volume on which the data set resides
Owner	Owner ID of the volume owner
Create date	Date the data set was created
Seq	Physical file sequence number

Example

Task: Create a list of all data sets that reside on volumes owned by OWN000 and that were created on or after March 14th, 1991.

Command:

```
RMM SEARCHDATASET OWNER(OWN000) SINCE(91073) LIMIT(*)
```

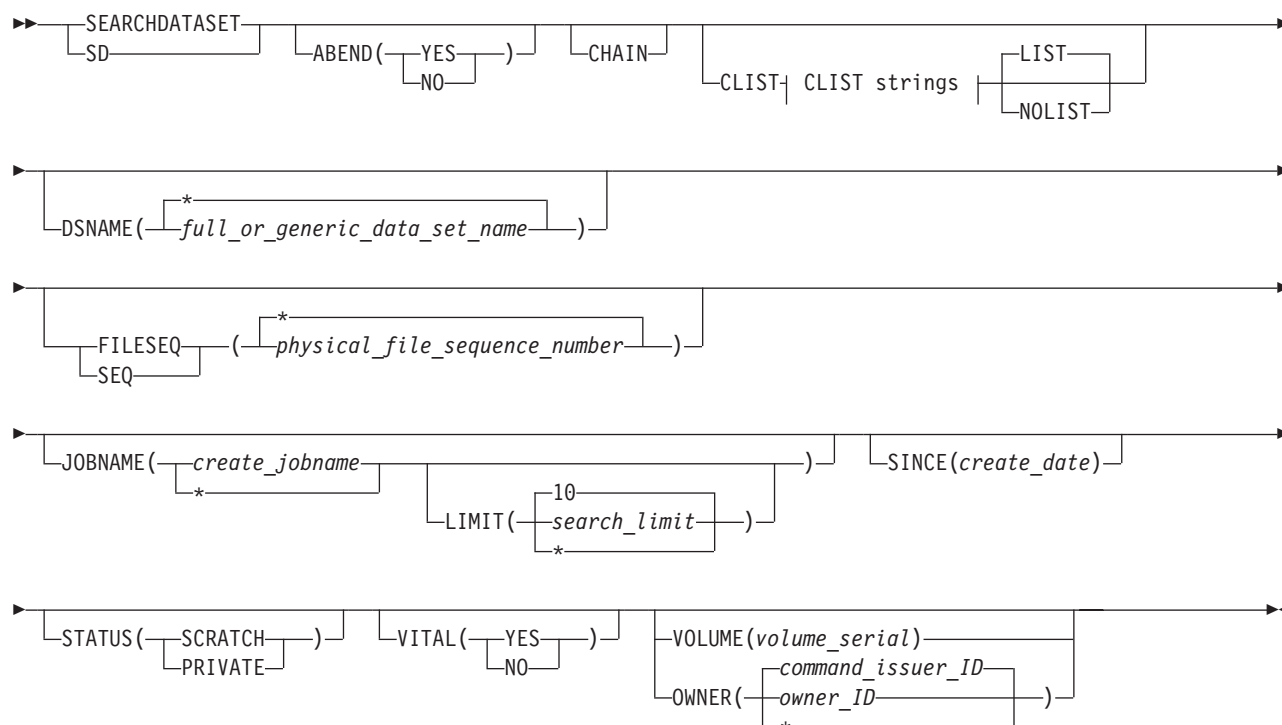
Output: DFSMSrmm displays information such as that shown in Figure 191:

Data set name	Volume	Owner	Create date	Seq
DATA.SET.FIVE	VOL001	OWN000	31/03/1991	1
DATA.SET.FOUR	VOL000	OWN000	21/03/1991	4
DATA.SET.SEVEN	VOL002	OWN000	16/03/1991	1
DATA.SET.TWO	VOL000	OWN000	17/03/1991	2

EDG3012I 4 ENTRIES LISTED

Figure 191. Sample SEARCHDATASET Output

Syntax



CLIST strings:

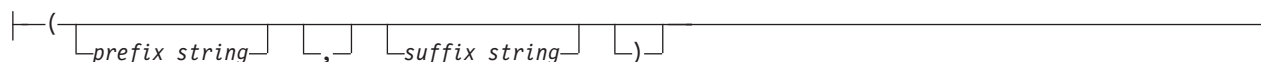


Figure 192. SEARCHDATASET Syntax Diagram

Operands

ABEND(YES,NO)

Specify to limit the search to volumes containing data sets that were closed as a result of ABEND processing.

CHAIN

Specify to search for all physical files in the same multi-volume data set. DFSMSrmm returns all the files in the set in volume sequence order starting from the first volume and file in the multi-volume data set. You must provide a data set name and a volume serial number of a physical file in the multi-volume data set. If you do not specify a file sequence number, DFSMSrmm uses the default file sequence value of 1. The value you specify must be the number of the physical file on the volume you specify for the search. The output might consist of files with numbers which are different from this number; a multi-volume data set can start on any physical file on a volume, but on each subsequent volume the file sequence number is 1.

This operand is optional and has no default.

CLIST(prefix_string,suffix_string)

Specify CLIST to create a data set of executable commands.

SEARCHDATASET Subcommand

For each data set that matches the search criteria, DFSMSRmm writes one or more records in a sequential data set. If RMMCLIST DD is pre-allocated, DFSMSRmm uses this data set for the CLIST output. If RMMCLIST is not pre-allocated, DFSMSRmm uses the data set called '*prefix*.EXEC.RMM.CLIST', where *prefix* is a TSO user PROFILE PREFIX value. If you do not change this value, it will be the same as your TSO user ID. If you have specified PROFILE NOPREFIX, DFSMSRmm gets the RACF user ID from the ACEE and uses it as the prefix as long as the RACF user ID is valid; otherwise, DFSMSRmm uses the job name. The data set uses blocked, variable-length records, with a logical record length of 255. If the length of the output record exceeds 255 characters, it is split into multiple records and DFSMSRmm adds a continuation character, +, to all but the last record. If the CLIST data set already exists, DFSMSRmm attempts to use it. If it is a partitioned data set, DFSMSRmm creates or reuses the member TEMPNAME.

Specify *prefix_string* and *suffix_string* to add subcommands and operands to the records in the file. DFSMSRmm returns the data as follows:

```
'prefix_string'dsname VOLUME(volume_serial_number)  
FILESEQ(physical_file_sequence_number)'suffix_string'
```

prefix_string and *suffix_string* can be any characters. You specify CLIST values in prefix and suffix text strings. The total length of each text string cannot exceed 255 characters. Enclose your text string in quotes if the string contains any separator characters, such as blanks and commas. Use a blank or a comma between the text strings when you specify both a prefix and a suffix value. Insert blanks in prefix and suffix values to prevent DFSMSRmm from concatenating the *prefix_string* with the data DFSMSRmm returns and the *suffix_string*.

If you do not specify a CLIST value, the file '*prefix*.EXEC.RMM.CLIST', contains only the name of the data set, a data set sequence number, and the volume serial number of the volume on which it resides for each record.

You can edit the CLIST to remove any data sets you do not want in the list, then you can run it at your convenience.

DSNAME(*,*full_or_generic_data_set_name*)

Specifies a data set name. Specify a fully-qualified data set name to list only those data sets that match the name exactly. In addition to normal data set naming conventions, you can use the following masking characters:

*** (asterisk)**

A single * represents a single qualifier of any number of characters.

A single * when used within a qualifier represents zero or more characters.

More than one single * can be used within a qualifier as long as a character precedes or follows the *.

.** represents zero or more qualifiers. At the end of the mask, it indicates to ignore any remaining characters.

** indicates to select all data sets.

% (percent sign)

A place holder for a single character.

SEARCHDATASET Subcommand

For example, you can specify DSNAME('USERID.**.CONF.*'). As another example, you can specify the fully-qualified data set name, USER.ALL.DATA or the generic data set name, USER.*.DATA.

You can specify an asterisk instead of a name to list all data sets that meet the other search criteria.

FILESEQ/SEQ(*,physical_file_sequence_number)

Specify to list data sets based on the relative position on a volume. The maximum allowable decimal value is 9999. Specify an * to list all data sets on the volume.

The default value is *.

JOBNAME(create_jobname,*)

Specifies the job name that created the data set.

A job name is one to eight alphanumeric characters or \$, #, or @. Job name must start with an alphabetic character, \$, #, or @. You cannot use a generic jobname. Any jobname you use must be specific.

If you do not specify JOBNAME, all data sets are listed.

If you specify JOBNAME(*), all data sets that match the specified data set name will be returned regardless of the job name. Data sets that do not have a job name are not listed.

LIMIT(search_limit,*)

Specifies how many entries DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify an asterisk to request a list of all entries matching the search criteria.

The default value is 10.

LIST

Specifies that DFSMSrmm produce a list when the CLIST operand is used.

LIST is mutually exclusive with the NOLIST operand. LIST is the default.

NOLIST

Specifies that DFSMSrmm not produce a list when the CLIST operand is used. DFSMSrmm produces only the CLIST output file.

NOLIST is mutually exclusive with the LIST operand. LIST is the default.

OWNER(user_ID,owner,*)

Specify to limit the search to volumes assigned to a specific owner ID. An owner ID is one to eight alphanumeric characters. The first character cannot be a number.

If you use an *, DFSMSrmm returns dataset information for all owners for data sets on both non-scratch and scratch volumes.

If you do not use the OWNER operand or you use a specific OWNER ID, DFSMSrmm returns data set information for data sets on non-scratch volumes only.

SEARCHDATASET Subcommand

The OWNER operand and the VOLUME operand are mutually exclusive. If you specify OWNER, you cannot specify the VOLUME operand. If you do not specify OWNER or VOLUME, DFSMSrmm uses the user ID of the command issuer as the default OWNER ID.

The default is the ID of the command issuer.

SINCE(*create_date*)

Specify to list only data sets created since the specified create date. Specify the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 1993/001. The slash is required.

If you do not specify the SINCE parameter, DFSMSrmm considers all data sets regardless of the date when they were created.

STATUS(SCRATCH,PRIVATE)

Specify to limit the search to data sets residing on scratch or private volumes. Specify STATUS(SCRATCH) to search for data sets on scratch volumes that have no owner ID assigned to them. Specify STATUS(PRIVATE) to search for data sets on master and user volumes.

VITAL(YES,NO)

Specify VITAL(YES) to limit the search to data sets that are retained by vital record specifications. If you do not specify this operand, retention by vital record specification is not part of the search criteria.

VOLUME(*volume_serial*)

Specify to limit the search to data sets residing on the indicated volume. A volume serial is one to six alphanumeric characters, or \$, #, or @, or special characters.

If you do not specify a volume serial, DFSMSrmm considers all volumes.

The VOLUME operand and the OWNER operands are mutually exclusive. If you specify VOLUME, DFSMSrmm does not consider the OWNER operand in the search.

Examples

Task: List all data sets on the volume with volume serial number, VOL000.

Command:

```
RMM SEARCHDATASET VOLUME(VOL000) LIMIT(*)
```

Output: DFSMSrmm displays a list such as the one shown in Figure 193 on page 367:

SEARCHDATASET Subcommand

Data set name	Volume	Owner	Create date	Seq
DATA.SET.ONE	VOL000	OWN000	19/02/1991	1
DATA.SET.TWO	VOL000	OWN000	01/03/1991	2
DATA.SET.THREE	VOL000	OWN000	09/02/1991	3
DATA.SET.FOUR	VOL000	OWN000	21/03/1991	4
EDG3012I 4	ENTRIES LISTED			

Figure 193. Sample SEARCHDATASET Output

Task: List all data sets on volumes belonging to OWN000 that are the first data set on the volume.

Command:

```
RMM SEARCHDATASET FILESEQ(1) OWNER(OWN000) LIMIT(*)
```

Output: DFSMSrmm displays a list such as the one shown in Figure 194:

Data set name	Volume	Owner	Create date	Seq
DATA.SET.FIVE	VOL001	OWN000	31/03/1991	1
DATA.SET.ONE	VOL000	OWN000	19/02/1991	1
DATA.SET.SEVEN	VOL002	OWN000	11/03/1991	1
EDG3012I 3	ENTRIES LISTED			

Figure 194. Sample SEARCHDATASET Output

Task: Generate a listing of all data sets belonging to WOODY.

Command:

```
RMM SEARCHDATASET DSNAME(*) OWNER(WOODY) LIMIT(*) -
CLIST('RMM LD ')
```

Output: DFSMSrmm displays a list such as the one shown in Figure 195:

Data set name	Volume	Owner	Create date	Seq
RMML01.SYSRES.BACKUP	999000	WOODY	26/02/1993	1
RMML01.SYSRES.BACKUP	999001	WOODY	26/02/1993	1
RMML01.SYSRES.BACKUP	999002	WOODY	26/02/1993	1
RMML01.SYSRES.BACKUP	999003	WOODY	26/02/1993	1
RMML01.SYSRES.BACKUP	999004	WOODY	26/02/1993	1
EDG3012I 5	ENTRIES LISTED			

Figure 195. Sample SEARCHDATASET Output

DFSMSrmm also creates a CLIST data set containing the records shown in Figure 196 on page 368:

SEARCHDATASET Subcommand

```
RMM LD 'RMML01.SYSRES.BACKUP' VOL(999000) FILESEQ(01)
RMM LD 'RMML01.SYSRES.BACKUP' VOL(999001) FILESEQ(01)
RMM LD 'RMML01.SYSRES.BACKUP' VOL(999002) FILESEQ(01)
RMM LD 'RMML01.SYSRES.BACKUP' VOL(999003) FILESEQ(01)
RMM LD 'RMML01.SYSRES.BACKUP' VOL(999004) FILESEQ(01)
```

Figure 196. Sample CLIST Data Set

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHPRODUCT: Creating a List of Software Products

Use the SEARCHPRODUCT subcommand as shown in Figure 198 on page 370 to create a list of software products defined to DFSMSrmm.

You can restrict how many software products DFSMSrmm displays by specifying the LIMIT operand. DFSMSrmm searches until it reaches your limit or until it lists all software products matching your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten products.

Table 33 shows the information DFSMSrmm returns for each software product in the list, in the order it is displayed:

Table 33. Information Returned By SEARCHPRODUCT

Table Field Name	Description
Number	Software product number
Level	Software product level
Product Name	Software product name
Feature Code	Software product feature code
Vols	Number of volumes associated with the software product
Volume	Volume serial number of the first volume where the software product resides

Example

Task: Create a list of all software products that have product numbers starting with PROD.

Command:

```
RMM SEARCHPRODUCT NUMBER(PROD*) LIMIT(*)
```

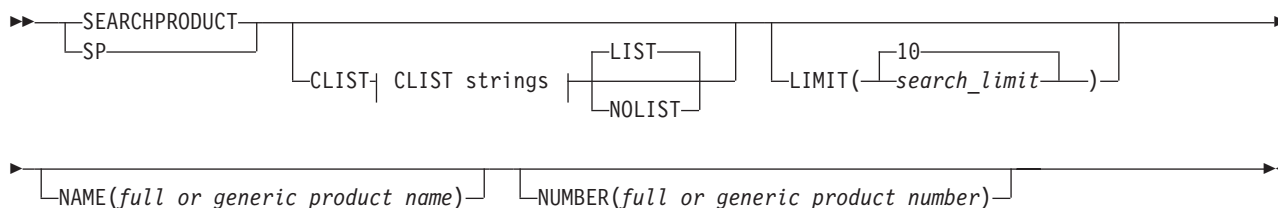
Output: DFSMSrmm displays a list such as the one shown in Figure 197:

Number	Level	Product Name	Feature Code	Vols	Volume
PROD01	V01R01M00	Product One	1234	5	VOL100
PROD02	V01R01M00	Product Two	3245	1	VOL800
PROD03	V01R01M00	Product Three	1059	1	VOL801
PROD04	V01R01M00	Product Four	9846	1	VOL802
PROD05	V01R01M00	Product Five	5647	1	VOL803
EDG3012I	5	ENTRIES LISTED			

Figure 197. Sample SEARCHPRODUCT Output

SEARCHPRODUCT Subcommand

Syntax



CLIST strings:

(*prefix_string*,*suffix_string*)

Figure 198. SEARCHPRODUCT Syntax Diagram

Operands

CLIST(*prefix_string*,*suffix_string*)

Specify CLIST to create a data set of executable commands.

For each software product that matches the search criteria, DFSMSrmm writes one or more records in a sequential data set. If RMMCLIST DD is pre-allocated, DFSMSrmm uses this data set for the CLIST output. If RMMCLIST is not pre-allocated, DFSMSrmm uses the data set called '*prefix*.EXEC.RMM.CLIST', where *prefix* is a TSO user PROFILE PREFIX value. If you do not change this value, it will be the same as your TSO user ID. If you have specified PROFILE NOPREFIX, DFSMSrmm gets the RACF user ID from the ACEE and uses it as the prefix as long as the RACF user ID is valid; otherwise, DFSMSrmm uses the job name. The data set uses blocked, variable-length records, with a logical record length of 255. If the length of the output record exceeds 255 characters, it is split into multiple records and DFSMSrmm adds a continuation character, +, to all but the last record. If the CLIST data set already exists, DFSMSrmm attempts to use it. If it is a partitioned data set, DFSMSrmm creates or reuses the member TEMPNAME.

Specify *prefix_string* and *suffix_string* to add subcommands and operands to the records in the file. DFSMSrmm returns the data as follows:

```
'prefix_string'product_number
LEVEL(product_level)'suffix_string'
```

prefix_string and *suffix_string* can be any characters. You specify CLIST values in prefix and suffix text strings. The total length of each text string cannot exceed 255 characters. Enclose your text string in quotes if the string contains any separator characters, such as blanks and commas. Use a blank or a comma between the text strings when you specify both a prefix and a suffix value. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the *prefix_string* with the data DFSMSrmm returns and the *suffix_string*.

If you do not specify a CLIST value, the file '*prefix*.EXEC.RMM.CLIST', contains only the software product number and version for each record.

You can edit the CLIST to remove any software products you do not want in the list, then you can run it at your convenience.

SEARCHPRODUCT Subcommand

LIMIT(*search_limit*,*)

Specify to limit how many entries DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify an asterisk to list all entries matching your search criteria.

The default value is 10.

LIST

Specifies that DFSMSrmm produce a list when the CLIST operand is used.

LIST is mutually exclusive with the NOLIST operand. LIST is the default.

NAME(*,*full_or_generic_software_product_name*)

Specifies a software product name. A full product name is one to thirty characters. A generic product name is 1 to 29 characters followed by an asterisk. Enclose the software product name in single quotes if it contains any special characters or blanks. Specify an asterisk to list software products regardless of name or number.

An asterisk is the default for NAME.

NOLIST

Specifies that DFSMSrmm not produce a list when the CLIST operand is used. DFSMSrmm produces only the CLIST output file.

NOLIST is mutually exclusive with the LIST operand. LIST is the default.

NUMBER(*,*full_or_generic_software_product_number*)

Specifies a software product number. A full software product number is one to eight characters. A generic software product number is one to seven characters followed by an asterisk. Enclose the value for NUMBER in single quotes if it contains any special characters or blanks. Specify an asterisk to list software products regardless of name or number.

An asterisk is the default value for NUMBER.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHRACK Subcommand

SEARCHRACK: Creating a List of Shelf Locations

Use the SEARCHRACK subcommand as shown in Figure 200 on page 373 to create a list of shelf locations defined in the removable media library. Shelf locations in the removable media library are called rack numbers.

You can restrict the number of rack numbers DFSMSrmm lists by specifying the LIMIT operand. DFSMSrmm searches until your limit is reached or until it lists all shelf locations matching your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten shelf locations.

Table 34 shows the information DFSMSrmm returns for each rack number in the list, in the order it is displayed:

Table 34. Information Returned by SEARCHRACK

Table Field Name	Description
Rack	Rack number
Medianame	Type of volume
Volume	Volume serial number
Status	Status of the shelf location (one of EMPTY, INUSE, or SCRATCH)
Location	Location where the volume resides

Example

Task: Create a list of fifteen scratch volumes to be pulled for use.

Command:

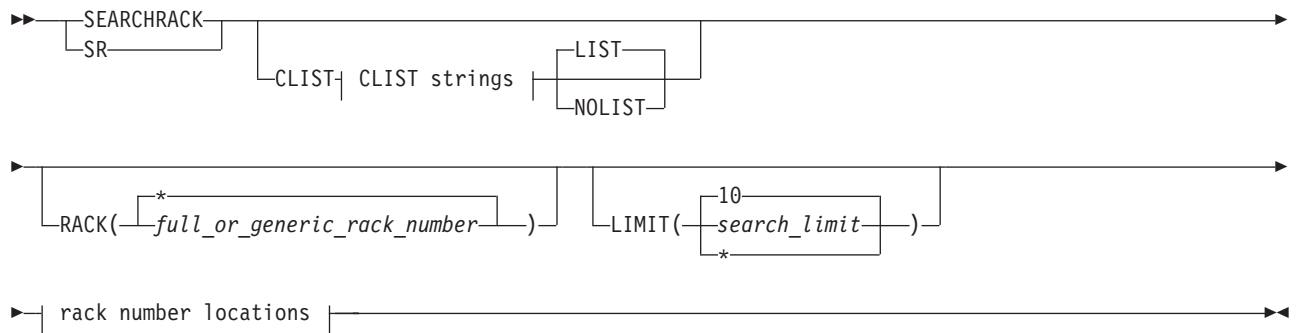
```
RMM SEARCHRACK POOL(RAC*) SCRATCH LIMIT(15)
```

Output: DFSMSrmm displays a list such as one shown in Figure 199:

Rack	Medianame	Volume	Status	Location
-----	-----	-----	-----	-----
RAC000	3480	SCR000	SCRATCH	SHELF
RAC001	3480	SCR001	SCRATCH	SHELF
RAC002	3480	SCR002	SCRATCH	SHELF
RAC003	3480	SCR003	SCRATCH	SHELF
RAC004	3480	SCR004	SCRATCH	SHELF
RAC005	3480	SCR005	SCRATCH	SHELF
RAC006	3480	SCR006	SCRATCH	SHELF
RAC007	3480	SCR007	SCRATCH	SHELF
RAC008	3480	SCR008	SCRATCH	SHELF
RAC009	3480	SCR009	SCRATCH	SHELF
RAC010	3480	SCR010	SCRATCH	SHELF
RAC011	3480	SCR011	SCRATCH	SHELF
RAC012	3480	SCR012	SCRATCH	SHELF
RAC013	3480	SCR013	SCRATCH	SHELF
RAC014	3480	SCR014	SCRATCH	SHELF
EDG3203I SEARCH COMPLETE - MORE ENTRIES MAY EXIST				
EDG3012I	15	ENTRIES LISTED		

Figure 199. Sample SEARCHRACK Output

Syntax



CLIST strings:

```
(prefix_string,suffix_string)
```

rack number locations:

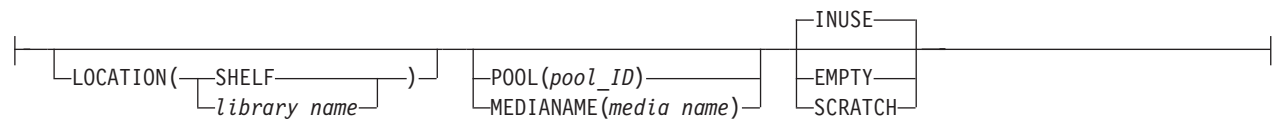


Figure 200. SEARCHRACK Syntax Diagram

Operands

CLIST(*prefix_string*,*suffix_string*)

Specify CLIST to create a data set of executable commands.

For each shelf location that matches the search criteria, DFSMSrmm writes one or more records in a sequential data set. If RMMCLIST DD is pre-allocated, DFSMSrmm uses this data set for the CLIST output. If RMMCLIST is not pre-allocated, DFSMSrmm uses the data set called '*prefix*.EXEC.RMM.CLIST', where *prefix* is a TSO user PROFILE PREFIX value. If you do not change this value, it will be the same as your TSO user ID. If you have specified PROFILE NOPREFIX, DFSMSrmm gets the RACF user ID from the ACEE and uses it as the prefix as long as the RACF user ID is valid; otherwise, DFSMSrmm uses the job name. The data set uses blocked, variable-length records, with a logical record length of 255. If the length of the output record exceeds 255 characters, it is split into multiple records and DFSMSrmm adds a continuation character, +, to all but the last record. If the CLIST data set already exists, DFSMSrmm attempts to use it. If it is a partitioned data set, DFSMSrmm creates or reuses the member TEMPNAME.

Specify *prefix_string* and *suffix_string* to add subcommands and operands to the records in the file. DFSMSrmm returns the data as follows:

```
'prefix_string'rack_number'suffix_string'
```

prefix_string and *suffix_string* can be any characters. You specify CLIST values in prefix and suffix strings. The total length of each text string cannot exceed 255 characters. Enclose your text string in quotes if the string contains any separator characters, such as blanks and commas. Use a blank or a

SEARCHRACK Subcommand

comma between the text strings when you specify both a prefix and a suffix value. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the *prefix_string* with the data DFSMSrmm returns and the *suffix_string*.

If you do not specify a CLIST value, the file '*prefix*.EXEC.RMM.CLIST', contains only the number of the shelf location for each record.

You can edit the CLIST to remove any shelf locations you do not want in the list, then you can run it at your convenience.

EMPTY

Specify to list only empty rack numbers in the removable media library. An empty rack number does not contain a volume and is available for use.

INUSE

Specify to list only those rack numbers in the removable media library that are in use.

INUSE is the default.

LIMIT(*search_limit*,*)

Specifies how many entries DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify an asterisk to request a list of all entries matching your search criteria.

The default value is 10.

LIST

Specifies that DFSMSrmm produce a list when the CLIST operand is used.

LIST is mutually exclusive with the NOLIST operand. LIST is the default.

LOCATION(SHELF,*library_name*)

Specify to limit the list to shelf locations in a specific library.

Specify SHELF to search for rack numbers in a non-system-managed library. Specify a library name to search for rack numbers in a specific system-managed library. A library name is one to eight alphanumeric characters.

MEDIANAME(*media_name*)

Specify to limit the list to shelf locations containing volumes with the same media name. Media names are defined by your installation and must be one to eight characters.

Use the LISTCONTROL subcommand with the VLPOOL operand to display media names defined for your location. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 333 for more information.

NOLIST

Specifies that DFSMSrmm not produce a list when the CLIST operand is used. DFSMSrmm produces only the CLIST output file.

NOLIST is mutually exclusive with the LIST operand. LIST is the default.

POOL(*pool_ID*)

Specifies a pool ID for a group of shelf locations from which DFSMSrmm lists rack numbers. A pool ID is one to five characters followed by an asterisk, and must be defined by your installation.

SEARCHRACK Subcommand

Specify POOL(*) to limit the search to rack numbers in the default scratch pool defined by your installation for your system. If you do not specify a pool ID, DFSMSrmm lists all rack numbers matching your search criteria, regardless of the pools with which they are associated.

RACK(*,*full_or_generic_rack_number*)

Specifies the rack number where you want DFSMSrmm to begin searching. A full rack number is six alphanumeric or national characters. A generic rack number is one to five alphanumeric or national characters followed by an asterisk. If you want DFSMSrmm to search through all the rack numbers defined to DFSMSrmm, use an asterisk as the rack number.

When using SEARCHRACK, DFSMSrmm returns rack numbers that match an operand that you specify or a default value set by DFSMSrmm. You can specify multiple operands but only INUSE, EMPTY, or SCRATCH have a default. RMM uses the default INUSE and returns only rack numbers that are in use. You must issue separate requests to list empty and scratch rack numbers. Use the DFSMSrmm ISPF dialog to obtain information about all the rack numbers.

* is the default.

SCRATCH

Specify to list only rack numbers associated with scratch volumes.

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHVOLUME Subcommand

SEARCHVOLUME: Creating a List of Volumes

Use the SEARCHVOLUME subcommand as shown in Figure 202 on page 378 to create a list of volumes matching the search criteria you specify. You can display lists of volumes based on ownership, assigned date, status, movement, action, pool or media name.

Use the ACTION operand to list those volumes with actions pending before they are released and returned to the scratch pool or an owner. You can specify the STATUS operand to tailor the list based on volume type.

Use the LIMIT operand to restrict how many volumes DFSMSrmm lists. DFSMSrmm searches until it reaches the limit you specify or until it lists all volumes matching your search criteria. DFSMSrmm lists a maximum of ten volumes if you do not specify a limit.

Table 35 shows the information DFSMSrmm returns for each volume in the list, in the order it is displayed:

Table 35. Information Returned by the RMM Searchvolume Subcommand

Table Field Name	Description
Volume	Volume serial number
Owner	Owner ID of the volume owner
Rack	Rack number
Assigned date	<ul style="list-style-type: none">• Date the volume is assigned to a user• Date the volume is in MASTER status when a non-specific volume is requested in a batch job• Date the volume is returned to scratch status
Expiration date	Date the volume is to be considered for release
Location	Location where the volume resides (one of SHELF, DISTANT, LOCAL, REMOTE, a <i>library_name</i> , or a <i>LOCDEF_location_name</i>)
Dsets	Number of data sets on the volume
St	Volume's status and availability, abbreviated as follows: M Master volume U User volume S Scratch volume I Scratch volume awaiting initialization E Scratch volume awaiting entry into a system-managed tape library O Volume is open for output V Vital record R Volume is pending release L Volume is on loan

SEARCHVOLUME Subcommand

Table 35. Information Returned by the RMM Searchvolume Subcommand (continued)

Table Field Name	Description
Act	Type of release action specified for the volume (E-erase, I-initialize, N-notify owner, O-return to owner, S-return to scratch, R-replace)
Dest.	Location where the volume is moving to. (One of SHELF, HOME, LOCAL, DISTANT, REMOTE, a <i>library_name</i> , or a <i>LOCDEF_location_name</i>)

Example

Task: Create a list of all volumes that belong to owner WEISSENB, and that were assigned after February 14th 1991.

Command:

```
RMM SEARCHVOLUME OWNER(WEISSENB) ACTION(ALL) SINCE(91055) LIMIT(*)
```

Output: DFSMSrmm displays a list such as shown in Figure 201:

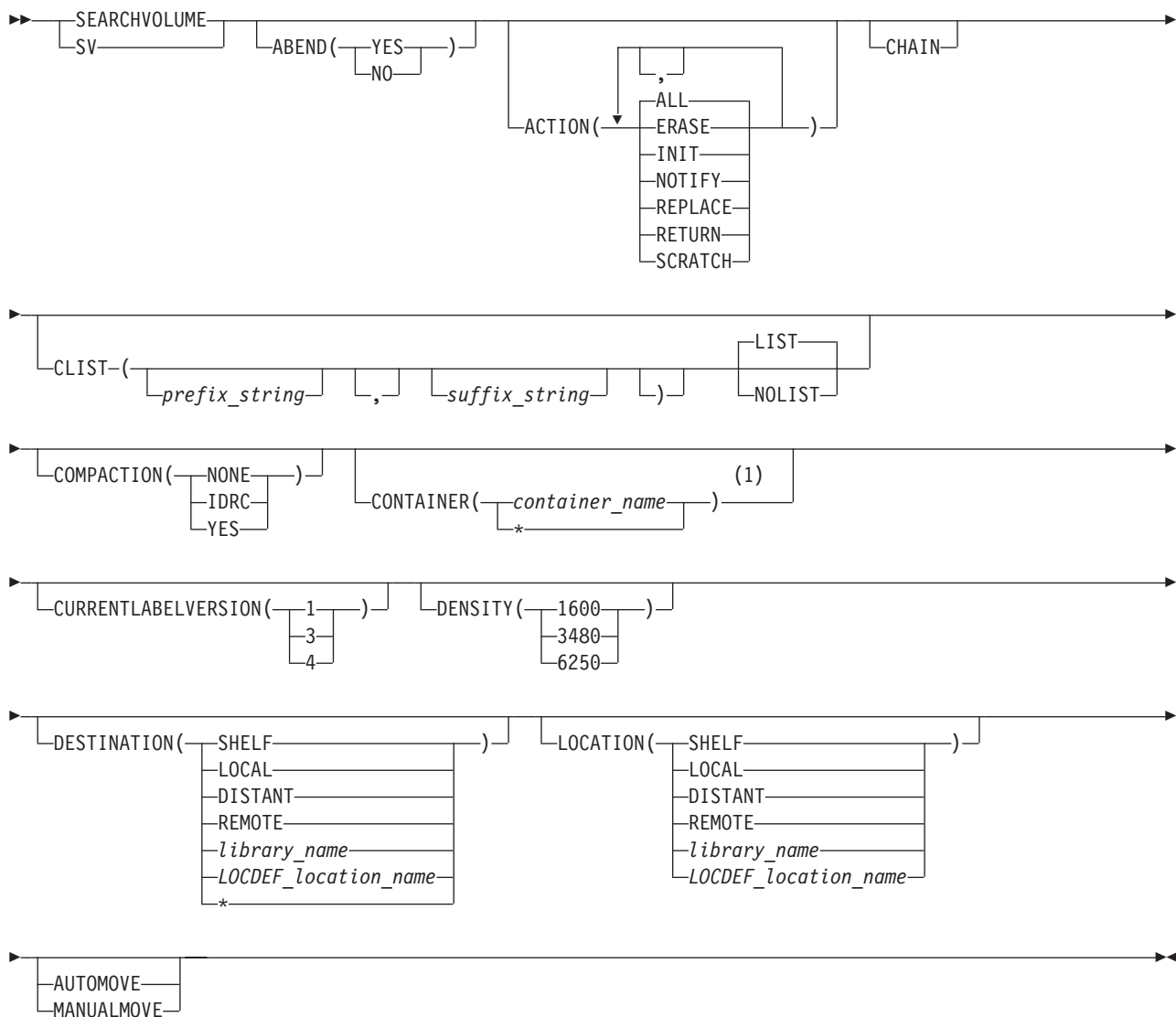
Volume	Owner	Rack	Assigned date	Expiration date	Location	Dsets	St	Act	Dest.
VOL600	WEISSENB	RAC500	06/11/1993	11/11/1993	SHELF	0		UR SI	

EDG3011I 1 ENTRY LISTED

Figure 201. Sample SEARCHVOLUME Output Listing Owned Volumes

SEARCHVOLUME Subcommand

Syntax



Notes:

1. Import/Export support is available with APAR OW36342 or OW36343.

Figure 202. SEARCHVOLUME Syntax Diagram Part 1

SEARCHVOLUME Subcommand

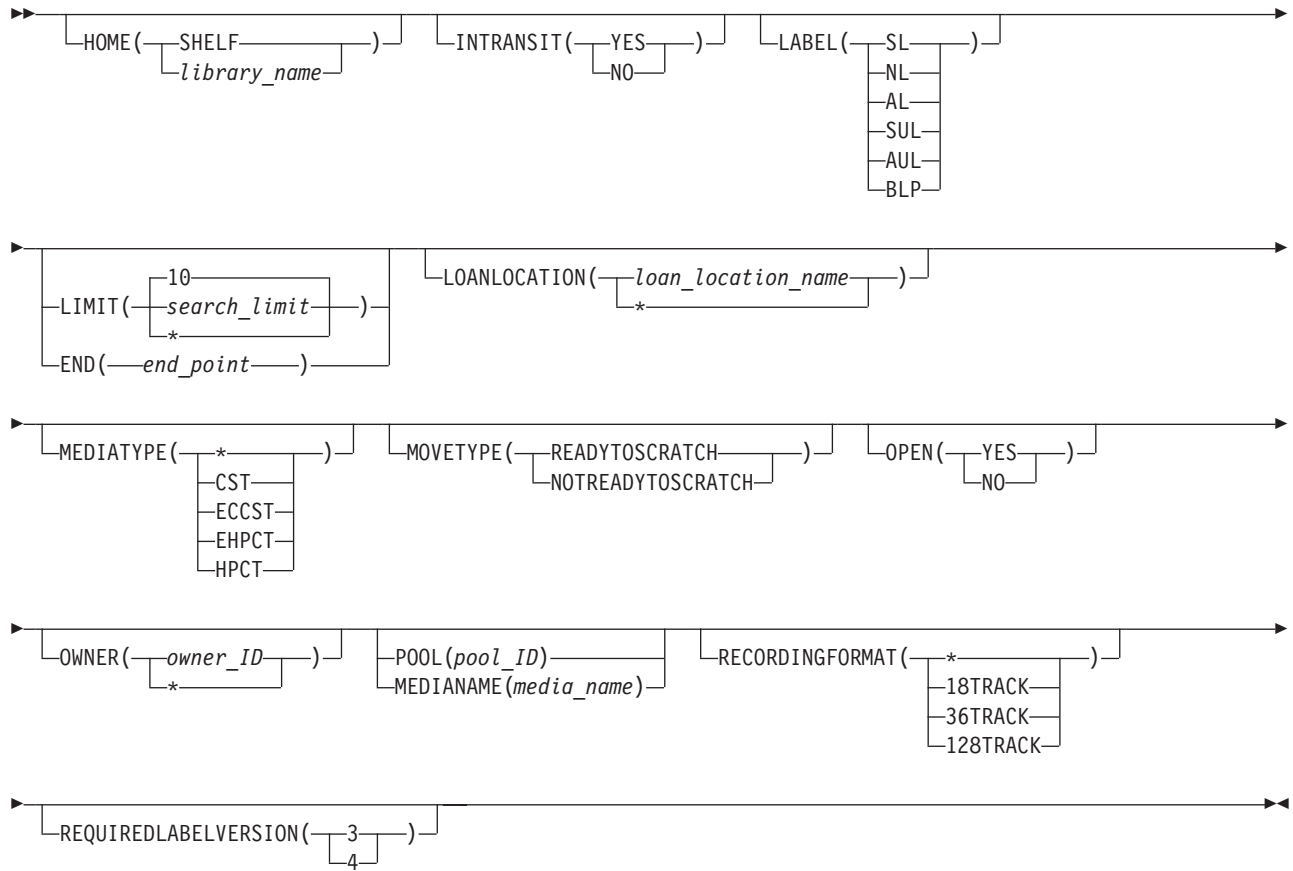
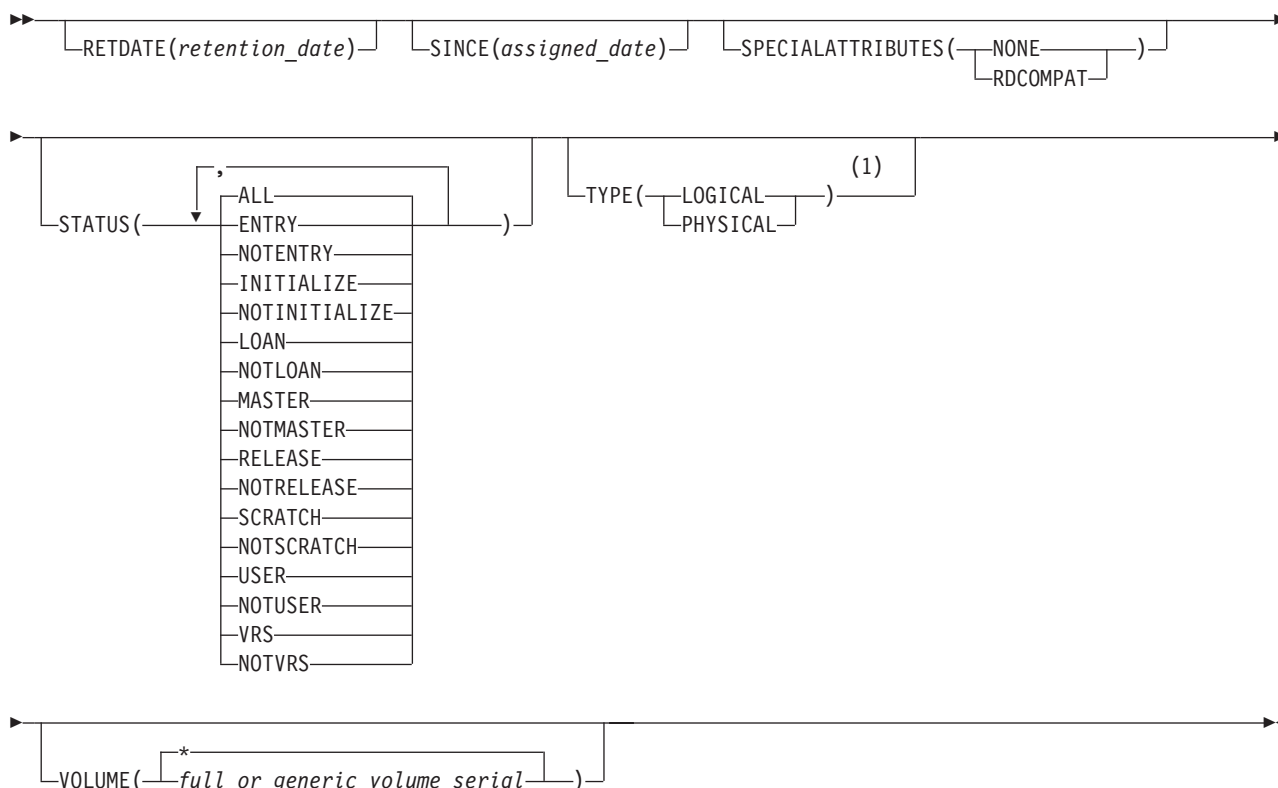


Figure 203. SEARCHVOLUME Syntax Diagram Part 2

SEARCHVOLUME Subcommand



Notes:

1. Import/Export support is available with APAR OW36342 or OW36343.

Figure 204. SEARCHVOLUME Syntax Diagram Part 3

Operands

ABEND(YES,NO)

Specify to limit the search to volumes containing a data set that was closed as a result of ABEND processing.

ACTION(ALL,ERASE,INIT,NOTIFY,REPLACE,RETURN,SCRATCH)

Specifies one or more actions DFSMSrmm uses as a search criteria. DFSMSrmm lists only volumes with the indicated pending release action. Specify a value of ALL, or one or more of the following, separated by commas:

ALL

To list all volumes with any pending action.

ERASE

To list only volumes that require erasing.

INIT

To list only volumes that require initialization.

RETURN

To list only volumes that should be returned to their owner.

REPLACE

To list only volumes that must be replaced by new volumes and returned to the scratch pool.

SEARCHVOLUME Subcommand

NOTIFY

To list only volumes for which owners must be notified.

SCRATCH

To list all volumes to be returned to scratch status.

AUTOMOVE

Specify to obtain a list of volumes that have a move mode of AUTOMOVE which indicates that DFSMSrmm automatic movement processing is in effect.

CHAIN

Specify this operand to search for all volumes in the same multi-volume data set. You must specify a specific volume serial number for any volume in the data set. DFSMSrmm retrieves the volume information from the control data set. DFSMSrmm uses the previous and next volume chain to return all volumes in the data set in volume sequence order, starting from the first volume in the data set.

This operand is optional and has no default.

CLIST(*prefix_string*,*suffix_string*)

Specify CLIST to create a data set of executable commands.

For each volume that matches the search criteria, DFSMSrmm writes one or more records in a sequential data set. If RMMCLIST DD is pre-allocated, DFSMSrmm uses this data set for the CLIST output. If RMMCLIST is not pre-allocated, DFSMSrmm uses the data set called '*prefix*.EXEC.RMM.CLIST', where *prefix* is a TSO user PROFILE PREFIX value. If you do not change this value, it will be the same as your TSO user ID. If you have specified PROFILE NOPREFIX, DFSMSrmm gets the RACF user ID from the ACEE and uses it as the prefix as long as the RACF user ID is valid; otherwise, DFSMSrmm uses the job name. The data set uses blocked, variable-length records, with a logical record length of 255. If the length of the output record exceeds 255 characters, it is split into multiple records and DFSMSrmm adds a continuation character, +, to all but the last record. If the CLIST data set already exists, DFSMSrmm attempts to use it. If it is a partitioned data set, DFSMSrmm creates or reuses the member TEMPNAME.

Specify *prefix_string* and *suffix_string* to add subcommands and operands to the records in the file. DFSMSrmm returns the data as follows:

```
'prefix_string'volume_serial_number'suffix_string'
```

prefix_string and *suffix_string* can be any characters. You specify CLIST values in prefix and suffix text strings. The total length of each text string cannot exceed 255 characters. Enclose your text string in quotes if the string contains any separator characters, such as blanks and commas. Use a blank or a comma between the text strings when you specify both a prefix and a suffix value. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the *prefix_string* with the data DFSMSrmm returns and the *suffix_string*.

If you do not specify a CLIST value, the file '*prefix*.EXEC.RMM.CLIST', contains only the volume serial number for each record.

You can edit the CLIST to remove any volumes you do not want in the list, then you can run it at your convenience.

SEARCHVOLUME Subcommand

COMPACTION(*,NONE,IDRC,YES)

Specifies the compaction technique used to record data on tape volumes. DFSMSrmm limits the list it returns to those volumes that match the specified value. Use one of the following:

***** The compaction is unknown or the volume is not a tape volume, and compaction does not apply. This is the default.

NONE No compaction was used to record data on the volume.

IDRC IDRC compaction which DFSMSrmm displays as a compaction value of YES.

YES The data on the master or user tape volumes defined to DFSMSrmm.

CONTAINER(*,container_name)¹⁸

Specifies to search for all volumes that are assigned to a container. Use * to select all volumes in any container or use a container name to select the volumes that are assigned to a specific container. If you do not specify the CONTAINER operand, DFSMSrmm selects all volumes that reside in and out of containers. CONTAINER can be any value that is defined to DFSMSrmm. The value can be any alphanumeric or special characters up to 16 characters in length.

CURRENTLABELVERSION(1,3, or 4)

Limits the output to those volumes that have the requested current label version.

There is no default.

DENSITY(*,1600,3480,6250)

Specifies to list volumes with a specific recording density. For a 3420 tape reel, you can specify DENSITY as 1600 or 6250. For a 3480 tape cartridge, specify a value of 3480. Specify an asterisk if you do not want to use density as a search criteria.

DESTINATION(*,SHELF,LOCAL,DISTANT,REMOTE,library_name,LOCDEF_location_name)

Specify to list volumes moving to a specific destination. If you specify DESTINATION(*), and you do not specify the LOCATION operand, DFSMSrmm lists all volumes that currently require moving.

Use the following values to identify the volume moves you want DFSMSrmm to list:

DISTANT

To list volumes moving to the DISTANT storage location

LOCAL

To list volumes moving to the LOCAL storage location

REMOTE

To list volumes moving to the REMOTE storage location.

LOCDEF_location_name

To list volumes moving to a storage location that was defined using the LOCDEF command.

You can enter any value as no checking is done against the current list of locations defined to DFSMSrmm.

18. Import/Export support is available with APAR OW36342 or OW36343.

SEARCHVOLUME Subcommand

library_name

To list volumes moving to a shelf location in a system-managed library

Library names must be one to eight alphanumeric characters, \$, #, or @, starting with a non-numeric character.

SHELF

To list volumes moving to shelf locations in a non-system-managed library

END(*end_point*)

Specify END as an alternative to the LIMIT operand to enable you to specify both the starting and ending point of the volume search. You do not need to know how many volumes are to be returned. The starting point is the volume serial number you provide and can be a specific volume serial number or a generic volume serial number. VOLUME(*) starts from the first volume. VOLUME(ABC*) starts with volume ABC or the next volume in collating sequence. If you specify the OWNER operand or use the default OWNER ID, DFSMSrmm starts with the first owned volume. END(*end_point*) identifies the last entry that DFSMSrmm returns. If the entry does not exist, DFSMSrmm does not return any entry with a volume serial number higher in collating sequence.

HOME(SHELF,*library_name*)

Specify to list volumes with the same home location name, whether they are currently stored in that location or not. The home location is where the volume should reside when not stored in a storage location. HOME can be a shelf location in a system-managed tape library or location SHELF.

A home location name is one to eight alphanumeric characters starting with a non-numeric character.

INTRANSIT(YES,NO)

Specify to list volumes that have started to move. If a volume has been ejected from a system-managed library, DFSMSrmm lists it as in transit. DFSMSrmm lists volumes in non-system-managed libraries as in transit as soon as inventory management identifies that a move is required.

Use the INTRANSIT operand together with LOCATION to limit the list to only those volumes residing in or moving from a specific location.

LABEL(SL,NL,AL,SUL,AUL,BLP)

Specify to list volumes of a specific volume type. You can specify any of the following:

SL

IBM standard labels

NL

No label

AL

ISO/ANSI labels

SUL

Standard user label

AUL

Both ISO/ANSI and user header or trailer labels

BLP

Bypass label processing

SEARCHVOLUME Subcommand

LIMIT(*search_limit*,*)

Specify to limit how many entries DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify an asterisk to request a list of all entries matching your search criteria.

The default value is 10.

LIST

Specifies that DFSMSrmm produce a list when the CLIST operand is used.

LIST is mutually exclusive with the NOLIST operand. LIST is the default.

LOANLOCATION(*loan_location_name*,*)

Specifies a loan location name. DFSMSrmm only lists volumes that reside in the specified loan location. Specify a specific loan location name to list volumes that reside in that particular location. A loan location is one to eight characters which must be enclosed in single quotes if it contains any blanks or special characters. Specify an * to list all volumes that reside in any loan location.

The LOANLOCATION operand has no default. If you do not specify LOANLOCATION, all volumes are listed whether or not they reside in a loan location.

LOCATION(SHELF,*library_name*,LOCDEF_*location_name*,

LOCAL,DISTANT,REMOTE)

Specify to list volumes residing in a specific location. Specify one of the following:

Specify SHELF to list volumes stored in shelf locations in a non-system-managed library.

Specify a *library_name* to list volumes stored in shelf locations in a specific system-managed library. A library name is one to eight alphanumeric characters starting with a non-numeric character.

Specify *LOCDEF_location_name* to list volumes moving to a storage location that was defined using the LOCDEF command.

You can enter any value as no checking is done against the current list of locations defined to DFSMSrmm.

Specify LOCAL, DISTANT, or REMOTE to list DFSMSrmm built-in storage locations.

Use the LOCATION operand together with INTRANSIT to limit the list to only those volumes residing in or moving from the specific location. Use the LOCATION operand together with HOME to limit the list to only those volumes residing in a specific location which have the same home location.

MANUALMOVE

Specify to obtain a list of volumes that have a move mode of MANUALMOVE.

MEDIANAME(*media_name*)

Specify to limit the list to volumes belonging to the same media name. Media name allows you to specify the type or shape of media. They are defined by your installation and must be one to eight characters.

SEARCHVOLUME Subcommand

Use the LISTCONTROL subcommand to display media names defined for your location. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 333 for more information.

MEDIATYPE(*,CST,ECCST,EHPCT,HPCT)

Specifies the volume's physical media type to limit the volume search. Use one of the following:

* the volume is not a cartridge. This is the default.

CST Cartridge System Tape

ECCST

Enhanced Capacity Cartridge System Tape

EHPCT

Reserved for Extended High Performance Cartridge Tape

HPCT High Performance Cartridge Tape

MOVETYPE(READYTOSCRATCH,NOTREADYTOSCRATCH)

Specify the READYTOSCRATCH operand to limit the list to volumes that are in ready to return to scratch status. Specify the NOTREADYTOSCRATCH operand to list volumes that are not pending release or are pending release with actions other than scratch.

NOLIST

Specifies that DFSMSrmm not produce a list when the CLIST operand is used. DFSMSrmm produces only the CLIST output file.

NOLIST is mutually exclusive with the LIST operand. LIST is the default.

OPEN(YES,NO)

Specify to limit the search to volumes with an open condition.

OWNER(owner,*)

Specifies an owner ID. DFSMSrmm only lists volumes belonging to the owner ID you specify. Specify a specific owner ID to list volumes belonging to that owner. Specify an asterisk to list all volumes that match the other search criteria regardless of their owner. An owner ID is one to eight alphanumeric characters or to six alphanumeric characters, \$, #, or @. The first character must not be a number. The default is your TSO user ID.

POOL(pool_ID)

Specifies the pool ID for a group of shelf locations DFSMSrmm uses to list volumes. DFSMSrmm only lists volumes associated with the pool you specify. A pool ID is one to five characters followed by an asterisk, and must be defined by your installation.

Use the LISTCONTROL subcommand with the VLPOOL operand to see the pool IDs defined for your installation.

RECORDINGFORMAT(*,18TRACK,36TRACK,128TRACK)

Specifies the basic recording format for tape volumes. DFSMSrmm limits the list it returns to those volumes that match the specified value.

* An asterisk indicates that the format is unknown or that the volume is not a tape volume. This is the default.

18TRACK

Data has been written to the volume in 18TRACKs.

SEARCHVOLUME Subcommand

36TRACK

Data has been written to the volume in 36TRACKs.

128TRACK

Data has been written to the volume in 128TRACKs. Recording format 128TRACK can be specified with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

REQUIREDLABELVERSION(3,4)

Limits the output to those volume entries that contain the requested required ISO/ANSI label version to be used in the VOL1 label for the volume.

There is no default.

RETDATE(*retention_date*)

Specifies that DFSMSrmm lists only MASTER and USER volumes which will expire up to and including the specified date. You can only specify dates. You cannot specify DFSMSrmm special date formats like CATRETPD or WHILECATALG. For volumes retained by a vital record specification, DFSMSrmm uses the retention date for the search. For volumes not retained by a vital record specification, DFSMSrmm uses the expiration date for the search. Specify the date in one of two forms:

- yyddd, where yy is the last two-digit number of the 20th century and ddd is the three-digit number for the day of the year, such as 98234.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such as 2003/200. A slash mark between the year and day is required.

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm defaults to the 20th century.

A volume's retention date is the latest of all the data sets on the volume. If you do not specify the RETDATE operand, DFSMSrmm searches all volumes, regardless of their retention date. See "Defining Retention Policies for Data Sets and Volumes" on page 103 for information about how DFSMSrmm calculates retention dates.

SPECIALATTRIBUTES(NONE,RDCOMPAT)

Specify to limit the list to volumes with special attributes associated with the tape volume.

RDCOMPAT

Specify to list only those volumes with the RDCOMPAT special attribute

NONE

Specify to list only those volumes with no special attributes

SINCE(*assigned_date*)

Specify to list volumes assigned to a user or volumes returned to scratch status since the *assigned_date*. Specify the year and day in one of two forms:

- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 93001.
- yyyy/ddd, where yyyy is the four-digit number for the year and ddd is the three-digit number for the day of the year, such 1993/001. The slash is required.

SEARCHVOLUME Subcommand

For dates in the year 2000 and or in the 21st century or higher, you can only use the yyyy/ddd format. If you use the yyddd format, DFSMSrmm defaults to the 20th century.

If you do not specify the SINCE operand DFSMSrmm searches all volumes, regardless of the date when they were assigned to a user.

STATUS(*status*)

Specify to list only volumes having the indicated status. *status* can be: ALL, ENTRY, INITIALIZE, MASTER, RELEASE, SCRATCH, USER, VRS, NOTENTRY, NOTINITIALIZE, NOTMASTER, NOTLOAN, NOTRELEASE, NOTSCRATCH, NOTUSER, or NOTVRS. Specify a value of ALL, or one or more of the following values separated by commas:

ALL

To list all volumes. ALL is the default.

ENTRY

To list volumes already defined to DFSMSrmm before being entered in an automated tape library dataserer for use as scratch volumes.

INITIALIZE

To list volumes waiting for initialization before becoming available for use as scratch volumes.

MASTER

To list volumes currently in master status.

RELEASE

To list volumes that are pending release and might require action, such as initializing.

SCRATCH

To list scratch volumes. This list does not include scratch volumes waiting for initialization or waiting for entry to an automated tape library dataserer before becoming available for use as scratch volumes.

USER

To list volumes currently in user status.

VRS

To list volumes retained by a vital record specification.

NOTENTRY

To exclude volumes already defined to DFSMSrmm from the list. These volumes are ones that have not yet been entered into an automated tape library dataserer for use as scratch volumes.

NOTINITIALIZE

To exclude volumes that must be initialized before they are available for use as scratch volumes.

NOTMASTER

To exclude volumes that are currently in master status.

NOTLOAN

To exclude volumes that are currently stored in loan locations.

NOTRELEASE

To exclude volumes that are pending release and might require a release action before they can be released.

SEARCHVOLUME Subcommand

NOTSCRATCH

To exclude scratch volumes from the list returned by the SEARCHVOLUME request. The list does not include scratch volumes waiting to be initialized or waiting for entry into an automated tape library dataserwer.

NOTUSER

To exclude volumes from the list returned by the SEARCHVOLUME request that are currently in USER status.

NOTVRS

To exclude volumes that are retained by vital record specifications.

TYPE(LOGICAL,PHYSICAL)¹⁹

Specify to limit the search to either logical or physical volumes. You can specify TYPE(LOGICAL) for VTS volumes only. When you specify the CLIST operand and TYPE(LOGICAL), DFSMSrmm returns more information in the output file. DFSMSrmm returns the first six characters of the container name, the logical volume serial number, and the status value. The status value can be:

- SCRATCH if the volume is in scratch status or ready to return to scratch with the SCRATCHIMMEDIATE release option set.
- INITIALIZE if the volume is in scratch status and contains no valid data.
- Blank if status is not available.

VOLUME(full_or_generic_volume_serial,*)

Specifies the serial number of the volume being searched. A full volume serial number is one to six alphanumeric characters, \$, #, or @, or special characters. A generic volume serial is one to five characters followed by an asterisk. Specify an asterisk to search all volumes matching the specified search criteria.

Example

Task: Search for all volumes moving from an automated tape library dataserwer with the library name ATL, and build a list containing eject commands for each volume.

Command:

```
RMM SEARCHVOLUME VOLUME(*) OWNER(*) LOCATION(ATL) -  
  DESTINATION(*) -  
  INTRANSIT(NO) LIMIT(*) CLIST('RMM CHANGEVOLUME ',' EJECT(BULK)')
```

Output: DFSMSrmm displays a list such as the one shown in Figure 205:

Volume	Owner	Rack	Assigned date	Expiration date	Location	Dsets	St	Act	Dest.
9990A5	ZWT01	9990A5	15/01/1993	20/01/1993	ATL	1	M		SHELF
9990A6	ZWT01	9990A6	15/01/1993	20/01/1993	ATL	1	M		SHELF
EDG3012I	2								

ENTRIES LISTED

Figure 205. Sample SEARCHVOLUME Output Listing Volumes to Eject

19. Import/Export support is available with APAR OW36342 or OW36343.

SEARCHVOLUME Subcommand

DFSMSrmm also creates a CLIST data set containing the records shown in Figure 206:

```
RMM CHANGEVOLUME 9990A5 EJECT(BULK)
RMM CHANGEVOLUME 9990A6 EJECT(BULK)
```

Figure 206. SEARCHVOLUME Output Using CLIST Operand

Task: Request a list of all the volumes that will expire up to and including the specified date 94300 which is October 27, 1994.

Command:

```
RMM SEARCHVOLUME OWNER(*) VOLUME(*) RETDATE(94300)
```

Output: DFSMSrmm displays a list such as the one in Figure 207:

Volume	Owner	Rack	Assigned date	Expiration date	Location	Dsets	St	Act	Dest.
KIM002	KIMBERLY	ABC002	06/04/1993	10/01/1994	SHELF	1		M	
KIM006	KIMBERLY	DAMW08	06/04/1993	10/01/1994	SHELF	1		M	
111001	LESLEY01	D65B35	06/18/1994	10/26/1994	SHELF	1		M	REMOTE
111013	LESLEY01	D65B36	06/18/1994	10/26/1994	SHELF	1		M	REMOTE

Figure 207. Sample SEARCHVOLUME Output Using RETDATE Operand

Task: Request a list of logical volumes with their containing stacked volume and status. When you specify TYPE(LOGICAL), DFSMSrmm returns the first six characters of the physical volume serial number, logical volume serial number, and the volume status.

Command:

```
RMM SEARCHVOLUME VOLUME(*) OWNER(*) DESTINATION(vts_name)-
TYPE(LOGICAL) CLIST
```

Output: DFSMSrmm displays a list such as the one in Figure 208:

```
CONT01,A00099
```

Figure 208. Sample SEARCHVOLUME Output Building a List of Logical Volumes

Task: Ensure that existing logical volumes are identified to DFSMSrmm as logical volumes.

Command:

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) LOCATION(vts_name)-
CLIST('RMM CHANGEVOLUME ', ' TYPE(LOGICAL)')
EXEC EXEC.RMM.CLIST
```

SEARCHVOLUME Subcommand

Output: DFSMSrmm displays a list such as the one in Figure 209:

```
RMM CHANGEVOLUME RFA050 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA051 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA052 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA053 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA054 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA055 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA056 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA057 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA058 TYPE(LOGICAL)
RMM CHANGEVOLUME RFA059 TYPE(LOGICAL)
```

Figure 209. Sample SEARCHVOLUME Output Identifying Logical Volumes to DFSMSrmm

Task: Move all volumes starting with volume serial number VOL from LOCATION(LIB1) to LOCATION(LIB2). LIB1 and LIB2 are manual or automated tape library data servers.

DFSMSrmm builds a CLIST data set. You can edit this data set to remove any volumes you do not want to move, and you can run it at your convenience. As each subcommand is processed, the volume is ejected from the library.

Command:

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) LOCATION(LIB1)-
  OWNER(*) INTRANSIT(NO)-
  CLIST('RMM CHANGEVOLUME ',' LOCATION(LIB2)')
```

Output: DFSMSrmm displays a list such as the one in Figure 210:

```
RMM CHANGEVOLUME RFA050 TYPE(LIB2)
RMM CHANGEVOLUME RFA051 TYPE(LIB2)
RMM CHANGEVOLUME RFA052 TYPE(LIB2)
RMM CHANGEVOLUME RFA053 TYPE(LIB2)
RMM CHANGEVOLUME RFA054 TYPE(LIB2)
RMM CHANGEVOLUME RFA055 TYPE(LIB2)
RMM CHANGEVOLUME RFA056 TYPE(LIB2)
RMM CHANGEVOLUME RFA057 TYPE(LIB2)
RMM CHANGEVOLUME RFA058 TYPE(LIB2)
RMM CHANGEVOLUME RFA059 TYPE(LIB2)
```

Figure 210. Moving Volumes from One Library to Another Library

Task: Confirm all volume moves between LOCATION(LIB1) and LOCATION(LIB2). LIB1 and LIB2 are manual or automated tape library data servers.

Command:

```
RMM SEARCHVOLUME VOLUME(*) LOCATION(LIB1) DESTINATION(LIB2) -
  INTRANSIT(Y) CLIST('RMM CHANGEVOLUME ',' MOVE') OWNER(*) LIMIT(*)
```

Output: DFSMSrmm displays a list such as the one in Figure 211:

```
RMM CHANGEVOLUME RFA199 MOVE
```

Figure 211. Confirming Volume Moves from One Library to Another Library

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHVRS Subcommand

SEARCHVRS: Creating a List of Vital Record Specifications

Use the SEARCHVRS subcommand as shown in Figure 213 on page 394 to create a list of vital record specifications.

Table 36. Creating Lists of Vital Record Specifications

To Request the List of	You Specify
Data set vital record specifications	A fully-qualified or generic data set name and optionally a job name
Volume vital record specifications	A specific or generic volume serial number
Name vital record specifications	A specific or generic vital record specification name

Use other operands to further limit your search. Specify the UNTILEXPIRED operand to restrict the search to those data sets which will be retained until the volume expiration date is reached. For example, you can request a list of chained vital record specifications or a list of vital record specifications owned by a specific owner. You can use the JOBNAME operand to limit the search to vital record specifications that match the job name mask.

Use the LIMIT operand to restrict how many vital record specifications DFSMSrmm lists. DFSMSrmm searches for vital record specifications until it reaches the limit you specify or until it lists all vital record specifications matching your search criteria. DFSMSrmm lists a maximum of ten vital record specifications if you do not specify the LIMIT operand.

Table 37 shows the information DFSMSrmm returns for each vital record specification in the list, in the order it is displayed:

Table 37. Information Returned by SEARCHVRS

Table Field Name	Description
Vital Record Specification	Volume serial number, data set name mask, or vital record specification name
Job name	Name of the job that created the data set
Type	Type of vital record specification (one of DSN, VOL or NAME)
Owner	Owner ID
Location	Location where the volume retained by the vital record specification resides
Next VRS	Name of next name vital record specification to which this vital record specification is linked

Example

Task: List all vital record specifications matching the data set name, MAXWEAD.**.

Command:

```
RMM SEARCHVRS DSNAME('MAXWEAD.**') OWNER(*) LIMIT(*)
```

Output: DFSMSrmm displays a list such as the one in Figure 212:

Vital Record Specification	Job name	Type	Location	Next VRS
-----	-----	-----	-----	-----
MAXWEAD.VRS.A.*	S181*	DSN	HOME	
MAXWEAD.VRS.A.*	S292*	DSN	REMOTE	
MAXWEAD.VRS.A.*	S313*	DSN	REMOTE	
MAXWEAD.VRS.A.*	S414*	DSN	REMOTE	
4 ENTRIES LISTED				

Figure 212. Sample SEARCHVRS Output

Syntax



Notes:

- Figure 213. SEARCHVRS Syntax Diagram

Operands**ANDVRS(*vrs_name*),***

Specifies that DFSMSrmm list only the vital record specifications that are chained using ANDVRS. Specify an asterisk to list all vital record specifications with an ANDVRS value.

BYDAYSCYCLE

Specifies that DFSMSrmm list only the vital record specifications that retain data sets using the BYDAYSCYCLE retention type.

CHAIN

Specifies that DFSMSrmm use the NEXTVRS or ANDVRS value specified for the vital record specification to search for a linked vital record specification. DFSMSrmm follows the chain and lists all linked vital record specifications in sequence.

You can specify CHAIN for a specific data set, volume, or name vital record specification. You can also specify CHAIN for a generic volume serial number or for a generic data set name mask as long as a vital record specification with that exact name has been defined to DFSMSrmm. When you use the CHAIN operand, DFSMSrmm ignores all other operands you specify except for the CLIST, LIST and NOLIST operands.

In some cases, DFSMSrmm cannot return all the vital record specifications in the chain. To obtain any remaining vital record specifications, reissue the search request, using the NEXTVRS value of the last item in the list as vital record specification name in your search request. For example, if G3 is the NEXTVRS or ANDVRS value for the last data set vital record specification in the returned list, specify

```
SEARCHVRS NAME(G3) CHAIN LIMIT(*)
```

to request that DFSMSrmm pick up the chain where it left off from the previous list.

CLIST(*prefix_string*,*suffix_string*)

Specify CLIST to create a data set of executable commands.

For each vital record specification that matches the search criteria, DFSMSrmm writes one or more records in a sequential data set. If RMMCLIST DD is pre-allocated, DFSMSrmm uses this data set for the CLIST output. If RMMCLIST is not pre-allocated, DFSMSrmm uses the data set called '*prefix*.EXEC.RMM.CLIST', where *prefix* is a TSO user PROFILE PREFIX value. If you do not change this value, it will be the same as your TSO user ID. If you have specified PROFILE NOPREFIX, DFSMSrmm gets the RACF user ID from the ACEE and uses it as the prefix as long as the RACF user ID is valid; otherwise, DFSMSrmm uses the job name. The data set uses blocked, variable-length records, with a logical record length of 255. If the length of the output record exceeds 255 characters, it is split into multiple records and DFSMSrmm adds a continuation character, +, to all but the last record. If the CLIST data set already exists, DFSMSrmm attempts to use it. If it is a partitioned data set, DFSMSrmm creates or reuses the member TEMPNAME.

Specify *prefix_string* and *suffix_string* to add subcommands and operands to the records in the file. For example, for a vital record specification, DFSMSrmm returns the data as follows:

```
'prefix_string'vrs_value'suffix_string'
```

SEARCHVRS Subcommand

prefix_string and *suffix_string* can be any characters. *vrs_value* can be a data set name, a data set name and a jobname, a volume, or a name. You specify CLIST values in prefix and suffix text strings. The total length of each text string cannot exceed 255 characters. Enclose your text string in quotes if the string contains any separator characters, such as blanks and commas. Use a blank or comma between the text strings when you specify both a prefix and a suffix value. Insert blanks in the prefix and suffix strings to prevent DFSMSrmm from concatenating the *prefix_string* with the data DFSMSrmm returns and the *suffix_string*.

If you do not specify a CLIST value, the file '*prefix*.EXEC.RMM.CLIST', contains only the data set name, volume serial number, or vital record specification name of the vital record specification for each record, depending on the type of vital record specification you are listing.

To avoid confusion between CLIST records, limit entries in the list to vital record specifications of the same type, such as data set, volume, or name.

You can edit the CLIST to remove any vital record specifications you do not want in the list, then you can run it at your convenience.

CYCLES

Specifies that DFSMSrmm list only those vital record specifications that retain data sets by cycles.

DAYS

Specifies that DFSMSrmm list only those vital record specifications that retain data sets by number of days.

DSNAME(*full_or_generic_data_set_name*,*)

Specifies a data set name. A data set name can be a fully qualified data set name or a generic data set name. In addition to normal data set naming conventions, you can use the following masking characters:

* (asterisk)

A single * represents a single qualifier of any number of characters.

A single * when used within a qualifier represents zero or more characters.

More than one single * can be used within a qualifier as long as a character precedes or follows the *.

.** represents zero or more qualifiers. At the end of the mask, it indicates to ignore any remaining characters.

** indicates to select all data sets.

% (percent sign)

A place holder for a single character.

~ (not sign)

A place holder for a single character. The ~ has special meaning in a data set name mask and is used to specify a pseudo-GDG data set name.

You can also specify an SMS management class name or a vital record specification management value. The name can be eight alphanumeric characters, beginning with an alphabetic character, and must follow standard MVS data set naming conventions. This name must be a single qualifier, and is already assigned by your installation.

SEARCHVRS Subcommand

For example, you can specify `DSNAME('M99000')`.

You can also use a data set name mask to list those vital record specifications that match to several management class names or vital record specification management values. For example, you could use the data set name mask `M9*` to request a list of vital record specifications covering any special dates in the range 98001 through 99366 that have been assigned a management class name or vital record specification management value. This data set name mask must be a single qualifier.

`DSNAME` is mutually exclusive with the `VOLUME` and `NAME` operands.

EXTRADAYS

Specifies that DFSMSrmm list only name vital record specifications that retain data sets by a number of extra days.

GDG

Specifies that DFSMSrmm list only data set vital record specifications for GDG based data set names.

JOBNAME(*jobname_mask*,*)

A job name is one to eight alphanumeric characters or \$, #, or @. The job name must start with an alphabetic character, \$, #, or @. You can specify a specific jobname or a jobname mask. Use % to match any one character and * to match any character string in the mask. If you do not specify `JOBNAME`, DFSMSrmm lists all matching vital record specifications with and without `JOBNAME`.

If you specify `JOBNAME(*)`, DFSMSrmm lists all the vital record specifications matching the specified `DSNAME` and defined with any `JOBNAME`. Data sets that do not have a job name are not listed. For example, the command:

```
RMM SEARCHVRS DSN('A.*') JOBNAME(*)
```

returns these vital record specifications:

- `DSN('A.B') JOBNAME(*)`
- `DSN('A.B') JOBNAME(A*)`
- `DSN('A.*') JOBNAME(*)`
- `DSN('A.A*') JOBNAME(ABC123)`

LIMIT(*search_limit*,*)

Specifies how many entries DFSMSrmm lists. The maximum allowable decimal value is 9999. Specify * to list all entries matching your search criteria.

The default value is 10.

LASTREFERENCEDAYS

Specifies that DFSMSrmm list only those vital record specifications that use the number of elapsed days since the data set was last read or written as a retention type.

LIST

Specifies that DFSMSrmm produce a list when the `CLIST` operand is used.

`LIST` is mutually exclusive with the `NOLIST` operand. `LIST` is the default.

LOCATION(*HOME*,*library_name*,*LOCAL*,*DISTANT*,*REMOTE*,*LOCDEF_location_name*)

Specify to list only those vital record specifications with the location indicated. *library_name* is any eight-character name meeting the system-managed library

SEARCHVRS Subcommand

naming convention restrictions. DFSMSrmm does not validate the location ID as a system-managed library. LOCDEF_location_name is any name up to eight characters long.

NAME(*full_or_generic_VRS_name*,*)

Specifies that DFSMSrmm list only the name vital record specifications matching the full or generic vital record specification name you specify. A full vital record specification name is a one to eight alphanumeric or national character name. A generic vital record specification name is zero to seven characters followed by an asterisk.

NAME is mutually exclusive with the DSNAME and VOLUME operands.

NEXTVRS(*VRS_name*,*)

Specifies that DFSMSrmm list only those vital record specifications that are chained to the vital record specification named *VRS_name*. Specify an asterisk to list all vital record specifications that contain either an ANDVRS or NEXTVRS value.

The default is to search regardless of vital record specification chaining.

NOGDG

Specifies that DFSMSrmm list only data set vital record specifications for NOGDG name data sets.

NOLIST

Specifies that DFSMSrmm not produce a list when the CLIST operand is used. DFSMSrmm produces only the CLIST output file.

NOLIST is mutually exclusive with the LIST operand. LIST is the default.

OWNER(*owner*,*)

Specifies that DFSMSrmm list only those vital record definitions belonging to the owner you specify. Specify an asterisk to list all vital record definitions regardless of their owner. An owner ID consists of one to eight alphanumeric characters, \$, #, or @. The first character cannot be a number. The default is your user ID.

RELEASE(EXPIRYDATEIGNORE,SCRATCHIMMEDIATE)

Specifies that DFSMSrmm restrict the search to data set vital record specifications where EXPIRYDATEIGNORE or SCRATCHIMMEDIATE have been coded.

UNTILEXPIRED

Specifies that DFSMSrmm restrict the search to vital recordspecifications where UNTILEXPIRED has been coded. The UNTILEXPIRED operand should be used in conjunction with DSNAME. Using UNTILEXPIRED with the VOLUME operand returns an empty set from the search.

VOLUME(*full_or_generic volume serial*,*)

Specifies a volume serial number. A full volume serial number is one to six alphanumeric characters, \$, #, or @, or special characters. A generic volume serial number is one to five characters followed by an asterisk. Specify an asterisk to include all volume vital record specifications in the search.

VOLUME is mutually exclusive with the DSNAME and NAME operands.

WHILECATALOG

Specifies that DFSMSrmm restrict the search to those vital record specifications specifying that DFSMSrmm retain a data set only as long as it is cataloged.

Examples

Task: Create a list of five volume vital record specifications beginning with the characters VOL.

Command:

```
RMM SEARCHVRS VOLUME(VOL*) LIMIT(5)
```

Output: DFSMSrmm displays a list such as the one in Figure 214:

Vital Record Specification	Job name	Type	Owner	Location	Next VRS
VOL300		VOL	OWN000	HOME	
VOL301		VOL	OWN000	HOME	
VOL302		VOL	OWN000	HOME	
VOL303		VOL	OWN000	HOME	
VOL304		VOL	OWN000	HOME	
EDG3012I 5					
ENTRIES LISTED					

Figure 214. Sample SEARCHVRS Output

Task: Create a list of vital record specifications in a chain, where the NEXTVRS value is VRS002.

Command:

```
RMM SEARCHVRS NAME(*) OWNER(*) NEXT(VRS002)
```

Output: DFSMSrmm displays a list such as Figure 215:

Vital Record Specification	Job name	Type	Owner	Location	Next VRS
VRS001		NAME	OWN000	HOME	VRS002
VRS003		NAME	OWN000	HOME	VRS002
VRS004		NAME	OWN000	HOME	VRS002
EDG3012I 3					
ENTRIES LISTED					

Figure 215. Sample SEARCHVRS Output

SEARCHVRS Subcommand

Return Codes

See “Chapter 13. DFSMSrmm Return and Reason Codes” on page 401 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

Chapter 13. DFSMSrmm Return and Reason Codes

This chapter lists return and reason codes issued by the RMM TSO subcommand, RMM, and a set of subcommands.

RMM TSO Subcommand Return Codes

Table 38 lists the return codes issued by DFSMSrmm.

Table 38. DFSMSrmm Return Codes

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. Incomplete or incorrect data and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

The command processor sets the return code values in Register 15 when the RMM TSO command or subcommands ends. You can find the value of the return code issued in the CLIST or REXX procedure environment.

For example, you can code statements, as shown in Figure 216, in a CLIST to test the return code issued by the LISTVOLUME subcommand:

```
PROC 0
RMM LISTVOLUME TEST01 ALL
SET RMMLC = &LASTCC
IF &RMMLC = 0 THEN +
    WRITE COMMAND OK
ELSE +
    WRITE COMMAND FAILED RC = &RMMLC
```

Figure 216. Testing RMM TSO LISTVOLUME Subcommand Example 1

You can also code REXX statements, as shown in Figure 217, to test the return code issued by the LISTVOLUME subcommand:

```
/*REXX*/
address "TSO" "RMM LISTVOLUME TEST01 ALL"
if rc = 0 then
    say "command ok"
else
    say "command failed rc=" rc
```

Figure 217. Testing RMM TSO LISTVOLUME Subcommand Example 2

RMM TSO Subcommand Reason Codes

DFSMSrmm issues reason codes that are only available in the REXX environment and only if command output is directed to REXX variables. The reason code is set into a fixed variable name by the command processor, EDG@RC. Where line mode output is used in the CLIST or the REXX procedure environment, the RMM TSO subcommand issues an explanatory message instead of setting a reason code. DFSMSrmm sets reason codes only when the return code is 4, 12, or 20.

DFSMSrmm sets reason codes for return code 20 only when DFSMSrmm processing determines an error other than value range or naming restrictions.

The following table lists the reason codes issued by DFSMSrmm when return code 4, 12, or 20 is issued. The table also lists related messages that can provide additional information about the error. DFSMSrmm messages can be found in *OS/390 MVS System Messages, Vol 2 (ASB-EWX)*.

For return code 20, an RMM message is not always issued as the TSO parse function provides the information messages for parse errors. The special reason codes for parse are only set when DFSMSrmm processing determines an error other than value, range, or naming restrictions. See Table 23 on page 221 for command abbreviations used in Table 39:

Table 39. DFSMSrmm Reason Codes

Return Code	Reason Code	Message Number	Issuing Command	Description
00	00	EDG3016	AV CV	Processing successful. DFSMSrmm returns a value for variable EDG@RCK when COUNT=1 or not used.
00	00	EDG3015	GV	Processing successful. DFSMSrmm returns a value for variable EDG@OWN and EDG@VOL when the RMM GETVOLUME request is successful.
04	02	EDG3203	SD	Search - more records might exist
04	02	EDG3203	SP	Search - more records might exist
04	02	EDG3203	SR	Search - more records might exist
04	02	EDG3203	SS	Search - more records might exist
04	02	EDG3203	SV	Search - more records might exist
04	04		SD	Search - fewer records returned than requested
04	04		SP	Search - fewer records returned than requested
04	04		SR	Search - fewer records returned than requested
04	04		SS	Search - fewer records returned than requested
04	04		SV	Search - fewer records returned than requested
04	06		All	Subsystem does not support function
04	06	EDG3312	SS	NEXTVRS does not exist for SS with CHAIN
04	08	EDG3010	SD	No data sets meet search criteria
04	08	EDG3010	SP	No software products meet search criteria
04	08	EDG3010	SR	No rack or bin numbers meet search criteria
04	08	EDG3010	SS	No vital record specifications meet search criteria
04	08	EDG3010	SV	No volumes meet search criteria
04	10		All SEARCH	Search - insufficient storage; more records might exist

Table 39. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
04	12	EDG3328	SD SV	Search chain- record in chain not found
12	00		All	I/O error on DFSMSrmm control data set
12	ALL	EDG3013	AV	DFSMSrmm issues values for variables EDG@CNT, EDG@RCK, and EDG@VOL when COUNT is greater than 1 and the command fails.
12	02	EDG3220	CO	Incorrect node and user ID combination
12	04	EDG3005	All	Issued when subcommand issued incorrectly
12	08	EDG3207	All	Subsystem does not exist
12	10	EDG3208	All	Disastrous error during subsystem processing
12	12	EDG3209	All	Logical error during subsystem processing
12	14	EDG3221	AD	Incorrect request for a scratch volume
12	14	EDG3221	CV	Incorrect request for a scratch volume
12	16	EDG3222	DO	Owner owns volumes but no new owner specified
12	18	EDG3017	AB AR DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the command fails and the COUNT operand is greater than 1.
12	18	EDG3018	AB AR DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the command fails and the COUNT operand is greater than 1.
12	18	EDG3200	AO	Owner already exists
12	18	EDG3200	AP	Software product already exists
12	18	EDG3200	AR	Rack or bin number already exists
12	18	EDG3200	AS	Vital record specification already exists
12	18	EDG3200	AV	Volume already exists
12	18	EDG3201	CD	Data set does not exist
12	18	EDG3201	CO	Owner does not exist
12	18	EDG3201	CP	Software product does not exist
12	18	EDG3201	CV	Volume does not exist
12	18	EDG3201	DD	Data set does not exist
12	18	EDG3201	DO	Owner does not exist
12	18	EDG3201	DP	Software product does not exist
12	18	EDG3201	DR	Rack or bin number does not exist
12	18	EDG3201	DS	Vital record specification does not exist
12	18	EDG3201	DV	Volume does not exist
12	18	EDG3201	LD	Data set does not exist
12	18	EDG3201	LO	Owner does not exist
12	18	EDG3201	LP	Software product does not exist
12	18	EDG3201	LR	Rack or bin number does not exist
12	18	EDG3201	LS	Vital record specification does not exist
12	18	EDG3201	SS	Vital record specification does not exist
12	18	EDG3201	LV	Volume does not exist

Table 39. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	20	EDG3204	All	I/O error on DFSMSrmm control data set
12	22	EDG3211	All	Subsystem abnormally ends due to incorrect data
12	24	EDG3212	All	Inventory management in progress - DFSMSrmm control data set cannot be changed
12	26	EDG3009	SD	Owner does not exist
12	26	EDG3009	DO	Owner does not exist
12	26	EDG3009	AP	Owner does not exist
12	26	EDG3009	CP	Owner does not exist
12	26	EDG3009	AS	Owner does not exist
12	26	EDG3009	AV	Owner does not exist
12	26	EDG3009	CV	Owner does not exist
12	26	EDG3009	GV	Owner does not exist
12	26	EDG3009	SV	Owner does not exist
12	28	EDG3223	AD	Volume does not exist
12	28	EDG3223	SV	Volume does not exist
12	28	EDG3223	SD	Volume does not exist
12	30	EDG3224	AD	Incorrect request for a DFSMSrmm recorded volume
12	30	EDG3224	CD	Incorrect request for a DFSMSrmm recorded volume
12	30	EDG3224	CV	Incorrect request for a DFSMSrmm recorded volume
12	30	EDG3224	DD	Incorrect request for a DFSMSrmm recorded volume
12	32	EDG3225	AD	A data set already exists at that position on the volume
12	34	EDG3226	AD	Previous data sets not defined on volume
12	36	EDG3227	SR	Incorrect rack number key in data area
12	36	EDG3228	SR	Incorrect bin number key in data area
12	38	EDG3229	DR	Rack number does not exist or is not empty
12	38	EDG3229	AV	Rack number does not exist or is not empty
12	38	EDG3229	CV	Rack number does not exist or is not empty
12	40	EDG3230	SR	Pool not defined
12	40	EDG3230	AV	Pool not defined
12	40	EDG3230	CV	Pool not defined
12	40	EDG3230	GV	Pool not defined
12	40	EDG3230	SV	Pool not defined
12	42	EDG3231	DV	Incorrect delete option in data area
12	44	EDG3232	DV	Volume is not a SCRATCH volume
12	46	EDG3233	DR	No empty bins in storage location
12	48	EDG3234	All	Incorrect date in data area
12	50	EDG3235	All	Incorrect time in data area
12	54	EDG3237	AV	Pool/unit mismatch
12	54	EDG3237	CV	Pool/unit mismatch

Table 39. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	56	EDG3238	AV	Pool is full
12	56	EDG3238	CV	Pool is full
12	58	EDG3239	AR	Incorrect storage location ID in data area
12	58	EDG3239	DR	Incorrect storage location ID in data area
12	58	EDG3239	LR	Incorrect storage location ID in data area
12	58	EDG3239	SR	Incorrect storage location ID in data area
12	60	EDG3240	AV	Unknown software product
12	60	EDG3240	CV	Unknown software product
12	62	EDG3241	CV	Unknown accessors in DELUSERS parameter
12	64	EDG3242	CV	Too many user IDs specified for ADDUSERS()
12	66	EDG3243	AV	Both Pool and rack specified in the data area
12	66	EDG3243	CV	Both pool and rack specified in the data area
12	68	EDG3017	AB AR DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the command fails and the COUNT operand is greater than 1.
12	68	EDG3018	AB AR DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the command fails and the COUNT operand is greater than 1.
12	68	EDG3244	AV	Count too big. Volser generated > 999999
12	68	EDG3244	AR	Rack or bin number greater than 999999
12	68	EDG3244	DR	Rack or bin number greater than 999999
12	70	EDG3017	AB AR DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the command fails and the COUNT operand is greater than 1.
12	70	EDG3018	AB AR DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the command fails and the COUNT operand is greater than 1.
12	70	EDG3245	AV	Count too big. Volume serial numeric suffix exhausted
12	70	EDG3245	AR	Count too big. Rack number numeric suffix exhausted
12	70	EDG3245	DR	Count too big. Rack number numeric suffix exhausted
12	72	EDG3246	AV	Previous volume already has a next volume
12	72	EDG3246	CV	Previous volume already has a next volume
12	74	EDG3247	DR	Bin not empty
12	76	EDG3248	DV	Volume is already a scratch volume
12	78	EDG3249	AV	Maximum volumes already assigned to software product
12	78	EDG3249	CV	Maximum volumes already assigned to software product
12	80	EDG3003	All	Severe error during TSO service routine
12	82	EDG3250	AV	No volume status supplied in data area
12	84	EDG3251	AV	More than 1 volume status supplied in data area
12	86	EDG3252	AV	Unknown previous volume
12	86	EDG3252	CV	Unknown previous volume
12	88	EDG3253	AV	Previous volume is a scratch volume

Table 39. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	88	EDG3253	CV	Previous volume is a scratch volume
12	90	EDG3254	CV	Volume has a next volume
12	92	EDG3255	GV	No default scratch pool of unit type defined
12	94	EDG3256	GV	No available volumes in pool
12	96	EDG3257	CV	Confirm action is not outstanding
12	98	EDG3258	CV	Confirm movement is not outstanding
12	100	EDG3259	AS	Count is too small
12	100	EDG3263	AS	COUNT is too small when the DELAY operand is used.
12	102	EDG3260	CV	Incorrect confirm release
12	104	EDG3008	All	Abnormal end in command processor
12	106	EDG3021	DO	New owner is the same as the old owner
12	108	EDG3022	AV, CV	Expiration date or retention period exceeds maximum
12	110	EDG3023	DV	Already pending release
12	114	EDG3265	AS	First STORENUMBER cannot be 99999
12	116	EDG3268	SD	Unable to open CLIST data set
12	116	EDG3268	SP	Unable to open CLIST data set
12	116	EDG3268	SR	Unable to open CLIST data set
12	116	EDG3268	SS	Unable to open CLIST data set
12	116	EDG3268	SV	Unable to open CLIST data set
12	118	EDG3269	SD	CLIST organization not sequential or partitioned
12	118	EDG3269	SP	CLIST organization not sequential or partitioned
12	118	EDG3269	SR	CLIST organization not sequential or partitioned
12	118	EDG3269	SS	CLIST organization not sequential or partitioned
12	118	EDG3269	SV	CLIST organization not sequential or partitioned
12	120	EDG3270	SD	CLIST data set too small
12	120	EDG3270	SP	CLIST data set too small
12	120	EDG3270	SR	CLIST data set too small
12	120	EDG3270	SS	CLIST data set too small
12	120	EDG3270	SV	CLIST data set too small
12	122	EDG3277	AV	Manual cartridge entry failed
12	122	EDG3277	CV	Manual cartridge entry failed
12	124	EDG3278	AV	Cannot override storage group. Current SG returned in EDG@CSG variable
12	124	EDG3278	CV	Cannot override storage group. Current SG returned in EDG@CSG variable
12	126	EDG3279	CV	Volume is not in a system-managed library
12	128	EDG3283	CV	Confirm move rejected until volume library resident
12	130	EDG3284	CV	RACK or POOL not allowed for a volume residing in a system-managed library
12	132	EDG3288	CV	Eject of a volume failed

Table 39. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	132	EDG3288	DV	Eject of a volume failed
12	134	EDG3289	CV	Manual cartridge entry failed during confirm move
12	136	EDG3290	CV	Confirm RETURN or REPLACE not accepted while volume is library resident
12	138	EDG3291	AV	Rack number and volser are not equal
12	138	EDG3291	CV	Rack number and volser are not equal
12	140	EDG3292	AV	Volume defined to SMS in different library. Current library returned in EDG@CLIB variable
12	140	EDG3292	CV	Volume defined to SMS in different library. Current library returned in EDG@CLIB variable
12	144	EDG3295	CV	LOCATION change rejected for volume that is already moving.
12	146	EDG3296	SS	CHAIN operand used but no exact match on vital record specification was found.
12	150	EDG3300	AV	Specified volume status conflicts with TCDB volume status
12	152	EDG3301	AV CV GV	Attempt to update volume status in TCDB failed.
12	154	EDG3302	CV	RACK or POOL not allowed for volume moving to a system-managed library
12	156	EDG3303	CV	RACK or POOL not allowed for volume with a home location of a system-managed library
12	158	EDG3280	CV	Move is not currently pending. CMOVE(from,to)
12	158	EDG3281	CV	Move is not currently pending. CMOVE(from,)
12	158	EDG3282	CV	Move is not currently pending. CMOVE(,to)
12	158	EDG3304	CV	Move is not currently pending. CMOVE(ALL,) or CMOVE(,ALL)
12	160	EDG3285	CV	Move is not currently confirmed. NOCMOVE(from,to)
12	160	EDG3305	CV	Move is not currently confirmed. NOCMOVE(from,)
12	160	EDG3306	CV	Move is not currently confirmed. NOCMOVE(,to)
12	160	EDG3307	CV	Move is not currently confirmed. NOCMOVE(ALL,) or CMOVE(,ALL)
12	162	EDG3286	CV	NOCRLSE failed, no actions pending.
12	164	EDG3287	CV	CRLSE failed, no actions pending.
12	166	EDG3308	SS	CHAIN operand specified without either DSNAME, NAME, or VOLUME operands.
12	168	EDG3309	CV	Confirm move rejected until volume ejected
12	170	EDG3310	CV DV	Volume location or destination not known on this system
12	172	EDG3311	AV CV DV	Update of TCDB failed
12	176	EDG3314	CV	Volume cannot be moved to a location because the media name is not eligible. Current media name returned in EDG@MEDN variable.
12	178	EDG3315	CV	Bin number does not exist or is not empty.
12	180	EDG3266	AS	COUNT is too small.

Table 39. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	182	EDG3267	AS	COUNT must equal STORENUMBER.
12	184	EDG3297	AS	STORENUMBER is missing.
12	186	EDG3325	SD SV	CHAIN specified with specific resource
12	188	EDG3327	AS	Data set name mask not suitable
12	190	EDG3326	AV CV	Current label version has been specified for a non-AL type tape volume
12	192	EDG3333	CV	HOME location value is not supported for a logical volume
12	194	EDG3329	CV DV	EJECT is not supported for a private logical volume
12	196	EDG3330	AV	TYPE and LOCATION are inconsistent
12	198	EDG3331	AV CV	RACK or POOL are not supported for a logical volume
12	200	EDG3332	CV	RACK number is not supported for a logical volume
20	0		All	Parse error for which we have no more specific reason.
20	2	EDG3106	AV	SECLEVEL value is not defined to DFSMSrmm
20	2	EDG3106	CV	SECLEVEL value is not defined to DFSMSrmm
20	2	EDG3106	LC	SECLEVEL value is not defined to DFSMSrmm
20	2	EDG3106	GV	SECLEVEL value is not defined to DFSMSrmm
20	2	EDG3106	CD	SECLEVEL value is not defined to DFSMSrmm
20	2	EDG3106	AD	SECLEVEL value is not defined to DFSMSrmm
20	4	EDG3107	AV	Do not use SECLEVEL as your installation has no security classes defined.
20	4	EDG3107	CV	Do not use SECLEVEL as your installation has no security classes defined.
20	4	EDG3107	LC	Do not use SECLEVEL as your installation has no security classes defined.
20	4	EDG3107	GV	Do not use SECLEVEL as your installation has no security classes defined.
20	4	EDG3107	CD	Do not use SECLEVEL as your installation has no security classes defined.
20	4	EDG3107	AD	Do not use SECLEVEL as your installation has no security classes defined.
20	6	EDG3272	AS	LOCATION operand specified a library that is not defined to DFSMS.
20	6	EDG3272	CV	LOCATION operand specified a library that is not defined to DFSMS.
20	6	EDG3272	GV	LOCATION operand specified a library that is not defined to DFSMS.
20	8	EDG3274	AS	LOCATION operand specified a library and libraries are not supported.
20	8	EDG3274	CV	LOCATION operand specified a library and libraries are not supported.
20	8	EDG3274	GV	LOCATION operand specified a library and libraries are not supported.

Table 39. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
20	10	EDG3273	AV	STORAGEGROUP operand value is not defined to DFSMS.
20	10	EDG3273	CV	STORAGEGROUP operand value is not defined to DFSMS.
20	12	EDG3275	AV	STORAGEGROUP operand is not supported.
20	12	EDG3275	CV	STORAGEGROUP operand is not supported.
20	14		AS	DSNAME value does not meet DFSMSrmm generic naming restrictions.
20	14		LS	DSNAME value does not meet DFSMSrmm generic naming restrictions.
20	14		DS	DSNAME value does not meet DFSMSrmm generic naming restrictions.
20	14		SS	DSNAME value does not meet DFSMSrmm generic naming restrictions.
20	14		SD	DSNAME value does not meet DFSMSrmm generic naming restrictions.
20	16	EDG3272	CV	HOME operand specified a library that is not defined to DFSMS.
20	18	EDG3274	CV	HOME operand specified a library and libraries are not supported.
20	20	EDG3276	AV	STATUS(VOLCAT) operand is not supported.
20	22	EDG3316	AB	Media name is not valid for the specified location.
20	24		AS	JOBNAME value does not meet DFSMSrmm generic naming restrictions.
20	24		DS	JOBNAME value does not meet DFSMSrmm generic naming restrictions.
20	24		LS	JOBNAME value does not meet DFSMSrmm generic naming restrictions.
20	24		SS	JOBNAME value does not meet DFSMSrmm generic naming restrictions.
20	26		AS	JOBNAME value does not meet DFSMSrmm naming restrictions.
20	26		CD	JOBNAME value does not meet DFSMSrmm naming restrictions.
20	26		SS	JOBNAME value does not meet DFSMSrmm naming restrictions.

Chapter 14. Creating REXX Execs

General-use programming interface

This chapter contains information you can use to create your own REXX execs or procedures to use with DFSMSrmm.

End of General-use programming interface

To get the TSO subcommands to return information as REXX variables, you must set the REXX variable SYSAUTH.EDGDATE to a valid abbreviation of a DATEFORM value.

All commands set EDG@RC if REXX special variable, RC, is 4, 12, or 20.

Some stem variables, such as EDG@VOL and EDG@DSN, use the stem value of 0 to indicate the number of items in the array. For example, if you issue the RMM SEARCHVOLUME subcommand, EDG@VOL.0 might contain 2, indicating two volumes met the search criteria. EDG@VOL.1 contains the first volume serial number and EDG@VOL.2 contains the second volume serial number.

Some variables like EDG@LDMN return information in a double stem variable. For example, if you issue the RMM LISTCONTROL LOCDEF subcommand, EDG@LDMN.1.0 variable contains the number of media names used for the first location. EDG@LDMN.1.1 contains the first media name, EDG@LDMN.1.2 the second media name. EDG@LDMN.2.0 variable contains the number of media names used for the second location, EDG@LDMN.2.1 contains the first media name, EDG@LDMN.2.2 the second media name.

TSO Subcommand Variables by Subcommand

Table 40 lists all the variables you can use in your REXX execs. The variables are listed in RMM subcommand order. "Sample REXX Execs" on page 426 provides examples of REXX execs using the variables.

Table 40. TSO Subcommand Variables by RMM Subcommand

Subcommand	Variable Name
ADDBIN	EDG@CNT EDG@LOC EDG@RC EDG@RCK
ADDDATASET	EDG@RC
ADDOWNER	EDG@RC
ADDPRODUCT	EDG@RC
ADDRACK	EDG@CNT EDG@LOC EDG@RC EDG@RCK
ADDVOLUME	EDG@CLIB EDG@CNT EDG@CSG EDG@FRC EDG@FRS EDG@RC EDG@RCK EDG@VOL
ADDVRS	EDG@RC
CHANGEDATASET	EDG@RC
CHANGEOWNER	EDG@RC
CHANGEPRODUCT	EDG@RC

Table 40. TSO Subcommand Variables by RMM Subcommand (continued)

Subcommand	Variable Name
CHANGEVOLUME	EDG@CLIB EDG@CSG EDG@FRC EDG@FRS EDG@MEDN EDG@RC EDG@RCK
DELETEBIN	EDG@CNT EDG@LOC EDG@RC EDG@RCK
DELETEDATASET	EDG@RC
DELETEOWNER	EDG@RC
DELETEPRODUCT	EDG@RC
DELETERACK	EDG@CNT EDG@LOC EDG@RC EDG@RCK
DELETEVOLUME	EDG@FRC EDG@FRS EDG@RC
DELETEVRS	EDG@NAME EDG@RC
GETVOLUME	EDG@FRC EDG@FRS EDG@OWN EDG@RC EDG@VOL
LISTBIN	EDG@LOC EDG@MEDN EDG@RC EDG@RCK EDG@RST EDG@VOL
LISTCONTROL ACTIONS	EDG@ACT ¹ EDG@AST ¹ EDG@RC
LISTCONTROL CNTL	EDG@AUD EDG@BDT EDG@BTM EDG@DBN EDG@CDS EDG@CSDT EDG@CSTM EDG@DDT EDG@DRP EDG@DTE EDG@DTM EDG@FBP EDG@FCSP EDG@FDB EDG@FEP EDG@FKP EDG@FLB EDG@FRB EDG@FRK EDG@FRP EDG@FSP EDG@FTP EDG@FVP EDG@FXP EDG@IPL EDG@JDS EDG@JRNU EDG@LBN EDG@LCT EDG@LRK EDG@MDS EDG@MDT EDG@MRP EDG@MTM EDG@MTP EDG@NOT EDG@OPM EDG@RBN EDG@RC EDG@RCF EDG@RDT EDG@RTM EDG@SAT EDG@SDT EDG@SID EDG@SLM EDG@SOSD EDG@SOSP EDG@SOST EDG@SSM EDG@STM EDG@VDT EDG@VTM EDG@V1 EDG@XDT EDG@XTM
LISTCONTROL LOCDEF	EDG@LDDF ¹ EDG@LDLC ¹ EDG@LDLT ¹ EDG@LDMN ³ EDG@LDMT ¹ EDG@LDPR ¹ EDG@RC
LISTCONTROL MOVES	EDG@MFR ¹ EDG@MST ¹ EDG@MTO ¹ EDG@MTY ¹

Table 40. TSO Subcommand Variables by RMM Subcommand (continued)

Subcommand	Variable Name
LISTCONTROL OPTION	EDG@ACCT EDG@AUD EDG@BLP EDG@BKPPEDG@CATS EDG@CDS EDG@CRP EDG@DRP EDG@DSPD EDG@DSPM EDG@DTE EDG@IPL EDG@JDS EDG@JRN EDG@LCT EDG@MDS EDG@MOP EDG@MRP EDG@MSGF EDG@OPM EDG@NOT EDG@PSFX EDG@RC EDG@RCF EDG@SID EDG@SLM EDG@SOSP EDG@SSM EDG@UNC EDG@VACT EDG@VCHG EDG@VMIN EDG@VRJ EDG@VRSL EDG@V1
LISTCONTROL SECCLS	EDG@CLS ² EDG@ERS ² EDG@MSG ² EDG@NME ² EDG@RC EDG@SEC ² EDG@SMF ²
LISTCONTROL SECLEVEL()	EDG@CLS EDG@DNM ¹ EDG@ERS EDG@MSG EDG@NME EDG@RC EDG@SEC EDG@SMF
LISTCONTROL VLPOOL	EDG@MEDN ¹ EDG@PDS ¹ EDG@PID ¹ EDG@PLN ¹ EDG@PRF ¹ EDG@PSN ¹ EDG@PTP ¹ EDG@RC EDG@XDC ¹
LISTCONTROL MNTMSG	EDG@MID ¹ EDG@OPL ¹ EDG@OVL ¹ EDG@RC EDG@SMI ¹
LISTCONTROL REJECT	EDG@GRK ¹ EDG@RC EDG@TAC ¹
LISTDATASET	EDG@ABND EDG@BLKC EDG@BLKS EDG@CDT EDG@CJBN EDG@CLS EDG@CTLGEDG@CTM EDG@DEV EDG@DLR EDG@DLW EDG@DSEQ EDG@DSN EDG@FILE EDG@LRCL EDG@MC EDG@NME EDG@OWN EDG@RC EDG@RCFM EDG@RTDT EDG@SC EDG@SG EDG@SYS EDG@VJBN EDG@VMV EDG@VNME EDG@VOL EDG@VRSR EDG@VSCD EDG@VSCN EDG@VTYP EDG@STEP EDG@DD EDG@2NME EDG@2SCD
LISTOWNER	EDG@AD1 EDG@AD2 EDG@AD3 EDG@DPT EDG@EMN EDG@EMU EDG@ETL EDG@FOR EDG@ITL EDG@OWN EDG@RC EDG@SUR EDG@VLN
LISTPRODUCT	EDG@FCD ¹ EDG@OWN EDG@PDSC EDG@PNME EDG@PNUM EDG@RC EDG@RCK ¹ EDG@VER EDG@VLN EDG@VOL ¹
LISTTRACK	EDG@LOC EDG@MEDN EDG@RC EDG@RCK EDG@RST EDG@VOL

Table 40. TSO Subcommand Variables by RMM Subcommand (continued)

Subcommand	Variable Name
LISTVOLUME VOL	EDG@ACN EDG@ACT EDG@ADT EDG@ATM EDG@AVL EDG@CDT EDG@CLS EDG@CTM EDG@CTNR EDG@DEN EDG@DESC EDG@DSN EDG@DSR EDG@JOB EDG@LBL EDG@LOAN EDG@LOC EDG@LVC EDG@LVN EDG@MEDA EDG@MEDC EDG@MEDR EDG@MEDT EDG@NME EDG@OCE EDG@OWN EDG@OXD EDG@PEND EDG@RC EDG@RCK EDG@RTDT EDG@SGN EDG@VOL EDG@VOLT EDG@VRSI EDG@VRXI EDG@VST EDG@XDT
LISTVOLUME ACCESS	EDG@ID1-12 EDG@LCID EDG@MVS EDG@OAC EDG@RC EDG@VAC EDG@VM
LISTVOLUME STATS	EDG@DEN EDG@DLR EDG@DLW EDG@DSC EDG@FCD EDG@LDEV EDG@NVL EDG@PNUM EDG@PRD EDG@PVL EDG@PWT EDG@RC EDG@SEQ EDG@TRD EDG@TWT EDG@MEDN EDG@USEC EDG@USEM EDG@VER
LISTVOLUME STORE	EDG@BIN EDG@BMN EDG@DEST EDG@HLOC EDG@INTR EDG@LOC EDG@LOCT EDG@MOVM EDG@NLOC EDG@OBMN EDG@OBN EDG@OLOC EDG@RC EDG@SDT
LISTVRS	EDG@DDT EDG@DESC EDG@DSN EDG@LOC EDG@NAME EDG@NVRS EDG@OWN EDG@PRTY EDG@RC EDG@RET EDG@RWC EDG@SC1 EDG@SC2 EDG@TYP EDG@UEX EDG@VANX EDG@VDD EDG@VJBN EDG@VOL EDG@VRC EDG@VRSI EDG@VRXI EDG@VSS
SEARCHBIN	EDG@MEDN ¹ EDG@RC EDG@RCK ¹ EDG@RST ¹ EDG@VOL ¹
SEARCHDATASET	EDG@CDT ¹ EDG@CDTJ ¹ EDG@CJBN ¹ EDG@CTM ¹ EDG@DSN ¹ EDG@FILE EDG@KEYF EDG@KEYT EDG@OWN ¹ EDG@RC EDG@TYPF EDG@TYPT EDG@VOL ¹
SEARCHPRODUCT	EDG@FCD ¹ EDG@OWN ¹ EDG@PNME ¹ EDG@PNUM ¹ EDG@RC EDG@VER ¹ EDG@VLN ¹ EDG@VOL ¹
SEARCHRACK	EDG@MEDN ¹ EDG@RC EDG@RCK ¹ EDG@RST ¹ EDG@VOL ¹

Table 40. TSO Subcommand Variables by RMM Subcommand (continued)

Subcommand	Variable Name
SEARCHVOLUME	EDG@ADT ¹ EDG@ADTJ ¹ EDG@AVL ¹ EDG@CTNR ¹ EDG@DEST ¹ EDG@DSC ¹ EDG@DSR EDG@HLOC ¹ EDG@INTR ¹ EDG@KEYF EDG@KEYT EDG@LBL ¹ EDG@LOAN ¹ EDG@LOC ¹ EDG@LVC ¹ EDG@LVN ¹ EDG@MEDA ¹ EDG@MEDC ¹ EDG@MEDN ¹ EDG@MEDR ¹ EDG@MEDT ¹ EDG@OWN ¹ EDG@PEND ¹ EDG@RC EDG@RCK ¹ EDG@RTDJ ¹ EDG@RTDT ¹ EDG@TYPF EDG@TYPT EDG@SEQ ¹ EDG@VOL ¹ EDG@VOLT ¹ EDG@VST ¹ EDG@XDT ¹ EDG@XDTJ ¹
SEARCHVRS	EDG@DDT ¹ EDG@DDTJ ¹ EDG@DSN ¹ EDG@LOC ¹ EDG@NVRS ¹ EDG@OWN ¹ EDG@PRTY ¹ EDG@RC EDG@RET ¹ EDG@RWC ¹ EDG@TYP ¹ EDG@UEX ¹ EDG@VANX ¹ EDG@VJBN ¹ EDG@VOL ¹ EDG@VRS ¹ EDG@VRSI ¹ EDG@VRXI ¹

Notes:

1. The variable is a stem variable.
2. The variable is a stem variable when you specify LISTCONTROL with SECCLS operand.
3. The variable is a double stem variable as described in “TSO Subcommand Variables by Subcommand” on page 411.

TSO Subcommand Variables by Name

Table 41 lists the variables you can use in your REXX execs by variable name. See Table 23 on page 221 for command abbreviations used in this section. “Sample REXX Execs” on page 426 provides examples of REXX execs using the variables.

Table 41. TSO Subcommand Variables by Name

Variable Name	Subcommands	Contents	Format
EDG@ABND	LD	ABEND while open	One of: YES or NO
EDG@ACCT	LC	Accounting information	J or S
EDG@ACN	LV	Account number	40 characters
EDG@ACT	LV	Actions to be performed on release	One of: SIEN, SIE, SIN, SEN, SI, SE, SN, S, RIEN, RIE, RIN, REN, RI, RE, RN, R, OIEN, OIE, OIN, OEN, OI, OE, ON, O
	LC ¹	Action name	One of: RETURN, REPLACE, INIT, ERASE, or NOTIFY
EDG@ADT	LV SV ¹	Assigned date	Date format
EDG@ADTJ	LV SV ¹	Assigned date	Julian date format
EDG@AD1	LO	Volume owner's address line 1	40 characters

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@AD2	LO	Volume owner's address line 2	40 characters
EDG@AD3	LO	Volume owner's address line 3	40 characters
EDG@AST	LC ¹	Action status	One of: Pending, Confirmed, Complete, or Unknown
EDG@ATM	LV SV	Assigned time	6 characters (hhmmss)
EDG@AUD	LC	SMF audit record number	Numeric: 0, 128-155 - 0 is no audit records written; 128-255 are the audit record numbers
EDG@AVL	LV SV ¹	Volume availability	15 characters, one of: On Loan, Pending Release, Open, or Vital record
EDG@BDT	LC	Last control data set backup date	Date format
EDG@BIN	LV	Bin number	Numeric: 0-999999 or 6 alphanumeric characters
EDG@BKPP	LC	Backup procedure name	1 to 8 alphanumeric characters
EDG@BLKC	LD SD	Number of data set blocks	Numeric: 1-999999
EDG@BLKS	LD SD	Size of data set blocks	Numeric: 1-32760
EDG@BLP	LC	BLP option	RMM or NORMM
EDG@BMN	LV	Bin number media name	8 characters
EDG@BTM	LC	Last control data set backup time	6 characters (hhmmss)
EDG@CATS	LC OPTION	CATSYSID value	One of: *, Notset, or Set
EDG@CDS	LC	Control data set identifier	8 characters
EDG@CDT	LD LV SD ¹	Volume or data set create date	Date format
EDG@CDTJ	SD ¹	Data set create date	Julian date format
EDG@CJBN	LD	Job name	8 characters
	SD ¹	Job name	8 characters
EDG@CLIB	AV ¹⁴ CV ¹⁴ DV ¹⁴	Current library name	8 characters
EDG@CLS	LC ² LD LV	Security classification description	32 characters
EDG@CNT	AB AR	Number of rack or bin numbers added	Numeric: 0-99999
	DB DR	Number of rack or bin numbers deleted	Numeric: 0-99999
	AV	Number of volumes added	Numeric: 0-99999
EDG@CRP	LC	CATRETPD retention period	Numeric: 0-9999
EDG@CSDT	LC CNTL	Catalog synchronize date	Date format
EDG@CSG	AV ¹³ CV ¹³	Current storage group name	8 characters

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@CSTM	LC CNTL	Catalog synchronize time	6 characters (hhmmss)
EDG@CTLG	LD	Catalog status	One of: UNKNOWN, YES, or NO
EDG@CTM	LD SD ¹	Data set create time	6 characters (hhmmss)
	LV	Volume create time	6 characters (hhmmss)
EDG@CTNR	LV SV ¹	Container in which the resource is stored (stacked volume)	16 characters
EDG@DBN	LC	Number of bin numbers in DISTANT storage location	Numeric: 0-999999
EDG@DC	LD	Data class name	8 characters
EDG@DD	LD	DD name in job that created the data set	8 characters
EDG@DDT	LC	Last inventory management run date	Date format
	LS SS ¹	Vital record specification deletion date	Date format
EDG@DDTJ	SS ¹	Vital record specification delete date	Julian date format
EDG@DEN	LV	Volume density	One of: 1600, 6250, 3480, or IDRC, or * (undefined)
EDG@DESC	LV	Volume description	30 characters
	LS	Vital record specification description	30 characters
EDG@DEST	LV SV ¹	Destination name	One of: SHELF, LOCAL, DISTANT, REMOTE a system-managed library name (8 characters) or blank
EDG@DEV	LD	Device address	3 or 4 hexadecimal characters
EDG@DLR	LV LD	Date data set on volume last read	Date format
EDG@DLW	LV LD	Date data set on volume last written to	Date format
EDG@DNM	LC ¹	Data set name mask	44 characters
EDG@DPT	LO	Owner's department	40 characters
EDG@DRP	LC	Default retention period	Numeric: 0-9999
EDG@DSC	LV SV ¹	Number of data sets on a volume	Numeric: 0-9999
EDG@DSEQ	LD	Data set sequence number	0-9999
EDG@DSN	LD LV LS ⁸ SD ¹ SS ¹	Data set name	44 characters
EDG@DSPD	LC	Disposition control DD name	8 characters
EDG@DSPM	LC	Disposition message prefix	8 characters
EDG@DSR	LV SV ¹	Data set recording	ON or OFF

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@DTE	LC	Installation date format	One of: A, E, I, or J
EDG@DTM	LC	Last inventory management run time	6 characters (hhmmss)
EDG@EMN	LO	Owner's node	8 characters
EDG@EMU	LO	Owner's user ID	8 characters
EDG@ERS	LC ²	Security classification erase option	Y or N
EDG@ETL	LO	Owner's external telephone number	20 characters
EDG@FBP	LC	Control data set 'backup in progress' flag	Y or N
EDG@FCD	LP ¹ LV SP ¹	Software product feature code	4 characters
EDG@FCSP	LC CNTL	Catalog Synchronize in progress	Y or N
EDG@FDB	LC	Number of free bin numbers in DISTANT storage location	Numeric: 0-999999
EDG@FEP	LC	Report extract processing in progress flag	Y or N
EDG@FILE	LD SD	Physical file sequence number	1-9999
EDG@FKP	LC	VRS processing in progress flag	Y or N
EDG@FLB	LC	Number of free bin numbers in LOCAL storage location	Numeric: 0-999999
EDG@FOR	LO	Owner's forename	20 characters
EDG@FRB	LC	Number of free bin numbers in REMOTE storage location	Numeric: 0-999999
EDG@FRC	AV CV DV GV	OAM return code	Numeric
EDG@FRK	LC	Number of free rack numbers in library	Numeric: 0-9999999999
EDG@FRP	LC	Control data set 'Restore in Progress' flag	Y or N
EDG@FRS	AV	OAM reason code	OAM return code I2
	CV	OAM reason code	OAM return code I2
	DV	OAM reason code	OAM return code I2
	GV	OAM reason code	OAM return code I2
EDG@FSP	LC	Storage location processing in progress flag	Y or N
EDG@FTP	LC	Satellite processing in progress flag	Y or N
EDG@FVP	LC	Control data set 'Verification in Progress' flag	Y or N
EDG@FXP	LC	Expiration processing in progress flag	Y or N

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@GRK	LC ¹	Generic rack number	6 characters
EDG@HLOC	LV SV ¹	Home location name	8 characters
EDG@ID1-12	LV	User IDs of authorized users	8 characters each
EDG@INTR	LV SV ¹	Volume intransit status	Y or N
EDG@IPL	LC	Date check required on IPL flag	Y or N
EDG@ITL	LO	Owner's internal telephone number	8 characters
EDG@JDS	LC	Journal name	44 characters
EDG@JOB	LV	Job name	8 characters
EDG@JRN	LC	JOURNALFULL parmlib operand value	Numeric 0-99
EDG@JRNU	LC	Journal percentage used	Numeric 0-100
EDG@KEYF	SD SV	Key from	Character
EDG@KEYT	SD SV	Key from	Character
EDG@LBL	LV	Volume label type	One of: AL, NL, or SL
	SV ¹	Volume label type	One of: AL, NL, SL, BLP, SUL, or AUL
EDG@LBN	LC	Number of bin numbers in LOCAL storage location	Numeric: 0-999999
EDG@LCID	LV	Last change user ID	8 characters
EDG@LCT	LC	Default number of lines per page for reports	Numeric: 10-999
EDG@LDDF	LC ¹	Existence of a location definition	YES or NO
EDG@LDEV	LV	Last drive	4 characters
EDG@LDLC	LC ^{1 3}	Location name	One of: SHELF, built-in or installation defined storage location name, or a system-managed library name (8 characters)
EDG@LDLT	LC ¹	Location type	One of: AUTO, MANUAL, STORE, or blank
EDG@LDMN	LC ³	Media name	8 characters
EDG@LDMT	LC ¹	Management type	BIN, NOBINS, or blank
EDG@LDPR	LC ¹	Location priority	Numeric: 0-9999
EDG@LOAN	LV SV ¹	Volume loan location	8 characters
EDG@LOC	AB AR DB DR	Location of volume, rack number or bin number	One of: LOCAL, DISTANT, REMOTE, or installation defined storage locations
	LB SB ¹	Location of volume, or bin number	Storage location name

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
	LR SR ¹	Location of volume, rack number	SHELF, or an 8 character name of a system-managed library name
	LS SS ¹	Location of volume, rack number or bin number	One of: DFSMSrmm built-in storage location name, installation defined storage location name, SHELF, or a system-managed library name
	LV SV ¹	Location of volume, rack number or bin number	One of: DFSMSrmm built-in storage location name, installation defined storage location name, SHELF, or a system-managed library name
EDG@LOCT	LV	Location type	One of: AUTO, MANUAL, STORE, or blank
EDG@LRCL	LD	Data set Logical Record Length (LRECL)	Numeric: 0-32761
EDG@LRK	LC	Number of LIBRARY rack numbers	Numeric: 0-999999999
EDG@MC	LD	Management class	8 characters, defined by your installation
EDG@LVC	LV SV ¹	Current label version	One of 1,3,4 or blank
EDG@LVN	LV SV ¹	Required label version	One of 3,4 or blank
EDG@MDS	LC	Control data set data set name	44 characters
EDG@MDT	LC	Control date set create date	Date format
EDG@MEDA	LV SV ¹	Tape special attributes	One of: NONE or RDCOMPAT
EDG@MEDC	LV SV ¹	Tape compaction type	One of: *, NONE, or YES
EDG@MEDN	CV LC ¹ LB LR LV SB ¹ SR ¹ SV ¹	Media name	8 characters
EDG@MEDR	LV SV ¹	Tape recording technique	One of: *, 18TRACK, 36TRACK, or 128TRACK
EDG@MEDT	LV SV ¹	Tape media type	One of: *, CST, ECCST, EHPCT,HPCT
EDG@MFR	LC ¹	Source location name	One of: DFSMSrmm built-in storage location name, installation defined storage location name, SHELF, or a system-managed library name
EDG@MID	LC ¹	Mount message ID	12 characters
EDG@MOVM	LV	Move mode	AUTO or MANUAL
EDG@MOP	LC	Master Overwrite	ADD, MATCH, LAST, or USER

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@MRP	LC	Maximum retention period	NOLIMIT or Numeric: 0-9999
EDG@MSG	LC ²	Security classification message option	Y or N
EDG@MSGF	LC	Case of message text	M or U
EDG@MST	LC ¹	Move status	One of: Pending, Confirmed, Complete, or Unknown
EDG@MTM	LC	Control data set create time	6 characters (hhmmss)
EDG@MTO	LC ¹	Target location name	One of: DFSMSrmm built-in storage location name, installation defined storage location name, SHELF, or a system-managed library name
EDG@MTY	LC ¹	Move type	RTS or NORTS
EDG@MVS	LV	MVS use flag	Y or N
EDG@NAME	DS LS ⁹	vital record specification name	8 characters
EDG@NLOC	LV	Required location name	8 characters
EDG@NME	LC ² LD LV	Security classification name	8 characters
EDG@NOT	LC	Notify volume owners	Y or N
EDG@NVL	LV	Next volume in sequence	1 to 6 alphanumeric characters or \$, @ and #
EDG@NVR	LS ¹⁶ SS ¹	Next vital record specification name	8 characters
EDG@OAC	LV	Owner access	One of: READ, UPDATE, or ALTER
EDG@OBMN	LV	Old bin number media name	8 characters
EDG@OBN	LV	Old bin number	Numeric: 0-999999 for DFSMSrmm built-in storage locations Alphanumeric: 6 characters for installation defined storage
EDG@OCE	LV	Volume information recorded at open, close or end-of-volume time	Y or N
EDG@OLOC	LV	Previous location name	8 characters
EDG@OPL	LC ¹	Position of the rack number or pool ID in mount message	Numeric: 1-999
EDG@OPM	LC	Operating mode	One of: M, R, W, or P
EDG@OVL	LC ¹	Position of the volume serial number in mount message	Numeric: 0-999
EDG@OWN	GV	Owner to whom volume has been assigned	Character 8

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
	AD LD	Owner of volume on which data set resides	Character 8
	AO LO	Owner	Character 8
	AP LP	Software product owner	Character 8
	AV LV	Volume owner	Character 8
	LS	Vital record specification owner	Character 8
	SD ¹	Owner of volume on which data set resides	Character 8
	SP ¹	Software product owner	Character 8
	SV ¹	Volume owner	Character 8
	SS ¹	Vital record specification owner	Character 8
EDG@OXD	LV	Original expiration date	Date format
EDG@PDS	LC ¹	Pool description	40 characters
EDG@PDSC	LP	Software product description	30 characters
EDG@PEND	LV SV ¹	Actions pending for volume	One of: SIEN, SIE, SIN, SEN, SI, SE, SN, S, RIEN, RIE, RIN, REN, RI, RE, RN, R, OIEN, OIE, OIN, OEN, OI, OE, ON, O
EDG@PID	LC ¹	Pool prefix	6 characters
EDG@PLN	LC ¹	Pool name	8 characters
EDG@PNME	LP SP ¹	Software product name	30 characters
EDG@PNUM	LP LV SP ¹	Software product number	8 characters
EDG@PRD	LV	Number of permanent read errors	Numeric: 0-99999
EDG@PRF	LC ¹	Pool definition RACF option	Y or N
EDG@PRTY	LS SS ¹	Priority	Numeric: 0-9999
EDG@PSFX	LC	Parmlib member suffix	2 characters
EDG@PSN	LC ¹	Pool definition system ID	8 characters
EDG@PTP	LC ¹	Pool definition pool type	R or S
EDG@PVL	LV	Previous volume in sequence	1 to 6 alphanumeric characters or \$, @ and #
EDG@PWT	LV	Number of permanent write errors	Numeric: 0-99999
EDG@RBN	LC	Number of bin numbers in REMOTE storage location	Numeric: 0-999999
EDG@RC	All subcommands ⁷	Reason code	Numeric:
EDG@RCF	LC	Installation RACF support	One of: N, P, or A
EDG@RCFM	LD	Data set record format (RECFM)	4 characters
EDG@RCK	AB AR AV ⁴ CV ⁵ DB DR LP ¹ LB LR LV SB ¹ SR ¹ SV ¹	Rack number	6 characters

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@RDT	LC	Date of last control data set report extract	Date format
EDG@RET	LS ¹¹	Retention type	One of: BYDAYC, CYCLES, DAYS, REFDAYS, VOLUMES, or XTRADAYS
	SS ¹	Retention type	One of: CYCLES, DAYS, REFDAYS, VOLUMES, or blank
EDG@RST	LB LR SB ¹ SR ¹	Rack or bin number status	One of: EMPTY, SCRATCH, or IN USE
EDG@RTDJ	SV ¹	Retention date	Julian date format
EDG@RTDT	LD LV SV ¹	Retention date	Calendar date, WHILECATLG, or CYCL/cccc
EDG@RTM	LC	Time of last control data set report extract	6 characters (hhmmss)
EDG@RWC	LS ⁸	Retain while cataloged	YES or NO
	SS ¹		YES, NO, or blank
EDG@SC	LD	Storage class name	8 characters
EDG@SC1	LS	Vital record specification first storage location days or cycles or volumes	Numeric: 1-99999
EDG@SDT	LV	Movement tracking date	Date format
EDG@SEC	LC ²	Security classification number	Numeric: 0-255
EDG@SEQ	LV SV ¹	Volume sequence number	Numeric: 1-255
EDG@SG	LD	Storage group name	8 characters
EDG@SGN	LV	Storage group name	8 characters
EDG@SID	LC	SMF system ID	8 characters
EDG@SLM	LC	MAXHOLD value	Numeric: 10-500
EDG@SMF	LC ²	Security classification SMF flag	Y or N
EDG@SMI	LC ¹	Start position of the message ID in mount message	Numeric: 0-999
EDG@SOSD	LC	Date last EDGXPROC was started	Date format
EDG@SOSP	LC	Name of short-on-scratch procedure	8 characters
EDG@SOST	LC	Time last EDGXPROC was started	6 characters (hhmmss)
EDG@SSM	LC	SMF security record number	Numeric: 0, 128-255; 0 is no security records written; 128-255 are the security record numbers
EDG@STEP	LD	Step name in job that created the data set	8 characters

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@SUR	LO	Owner's surname	20 characters
EDG@SYS	LD	SMF system ID for system on which data set was written	8 characters
EDG@TAC	LC ¹	Reject prefix type	One of: READONLY or NONE
EDG@TRD	LV	Number of temporary read errors	Numeric: 0-99999
EDG@TWT	LV	Number of temporary write errors	Numeric: 0-99999
EDG@TYP	LS	Vital record specification type	One of: GDG, PSEUDO-GDG, DSNNAME, VOLUME, or NAME
	SS ¹		One of: GDG, PGDG, DSN, VOL, or NAME
EDG@TYPF	SD SV	Key from	Character
EDG@TYPT	SD SV	Key from	Character
EDG@UEX	LS ⁸	Retain until expired	YES or NO
	SS ¹		YES, NO, or blank
EDG@UNC	LC	Status of uncatalog processing	One of: Y, N, or S
EDG@USEC	LV	Volume use count	Numeric: 0-99999
EDG@USEM	LV	Volume usage (KB)	Numeric: 0-9999999999
EDG@VAC	LV	Volume access	One of: READ, UPDATE, or NONE
EDG@VACT	LD	VRSMIN action	One of: FAIL, INFO, WARN
EDG@VANX	LS SS ¹	Next vital record specification type	One of: NEXT, AND, or blank
EDG@VCHG	LC	VRSCCHANGE value	One of: INFO, VERIFY
EDG@VDD	LS	Vital record specification delay days ¹¹	Numeric: 0-99
EDG@VDT	LC	Date of last inventory management vital record processing	Date format
EDG@VER	LP LV SP ¹	Software product version	vrrmm
EDG@VJBN	LD	VRS job name mask	8 characters
	LS ⁸	Job name	8 characters
	SS ¹	Job name	8 characters ¹²
EDG@VLN	LO	Number of owned volumes	Numeric: 0-4294867296
	LP SP ¹	Number of software product volumes	Numeric: 0-999
EDG@VM	LV	VM use flag	Y or N
EDG@VMIN	LC	VRSMIN count value	A number
EDG@VMV	LD	Vital record specification management value	8 characters, defined by your installation

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@VNME	LD	Vital record specification name	44 characters
EDG@VOL	AV GV LD LP ¹ LB LR LV LS ¹⁰ SB ¹ SD ¹ SP ¹ SR ¹ SS ¹ SV ¹	Volume serial number	1 to 6 alphanumeric characters or \$, @ and #
EDG@VOLT	LV SV ¹	Volume type	One of: LOGICAL, PHYSICAL
EDG@VRC	LS ¹¹	Vital record count	Numeric: 1-99999
EDG@VRJ	LC	Vital record specification job name	Numeric: 1 or 2
EDG@VRS	SS ¹	Vital record specification	Either a data set name (up to 44 characters), a vital record specification name (up to 8 characters) or a volume serial number (up to 6 characters)
EDG@VRSI	LV LS SS ¹	SCRATCHIMMEDIATE release option	One of: YES or NO
EDG@VRSL	LC OPTION	VRSEL value	One of: OLD or NEW
EDG@VRXI	LV LS SS ¹	EXPIRYDATEIGNORE release option	One of: YES or NO
EDG@VRSR	LD	Data set vital record status	YES or NO
EDG@VSCD	LD	Primary vital record specification subchain date	Date format
EDG@VSCN	LD	Primary vital record specification subchain name	8 characters
EDG@VSS	LS ¹¹	Vital record start number	Numeric: 1-9
EDG@VST	LV SV ¹	Volume status	One of: MASTER, SCRATCH, USER, INIT, or ENTRY
EDG@VTM	LC	Time of last inventory management vital record processing	6 characters (hhmmss)
EDG@VTYP	LD	Primary vital record specification type	DATASET, SMSMC, VRSMV, DSN/MV, or DSN/MC
EDG@V1	LC	Calling TLCS by DFSMSrmm	Y or N
EDG@XDC	LC ¹	Expiration date check	One of: Y, N, or O
EDG@XDT	LC	Date of last inventory management expiration processing	Date format
	LV SV ¹	Volume expiration date	Date format
EDG@XDTJ	SV ¹	Volume expiration date	Julian date format
EDG@XTM	LC	Time of last inventory management expiration processing	6 characters (hhmmss)

Table 41. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@2JBN	LD	Secondary vital record specification jobname mask	Date format
EDG@2NME	LD	Secondary vital record specification mask	8 characters
EDG@2SCD	LD	Secondary vital record specification subchain date	Date format
EDG@2SCN	LD	Secondary vital record specification subchain name	8 characters

Notes:

1. The variable is a stem variable.
2. The variable is a stem variable when you specify LISTCONTROL with SECCLS operand.
3. The variable is a double stem variable as described in "TSO Subcommand Variables by Subcommand" on page 411.
4. The variable is set when you do not specify the RACK operand and the COUNT operand is either 1 or not specified.
5. The variable is set only when you specify the POOL operand.
6. The variable is set when an unexpected return code from a TSO service routine is detected.
7. EDG@RC is set if the return code in the REXX special variable is 4, 12, or 20.
8. The variable is set when listing a data set vital record specification.
9. The variable is set when listing a name vital record specification.
10. The variable is set when listing a volume vital record specification.
11. The variable is not set for a name vital record specification.
12. The variable is blank for a name or volume vital record specification.
13. The variable is set if the return code is 12 and the reason code is 124.
14. The variable is set if the return code is 12 and the reason code is 140.
15. The variable is set if EDG@LOC=BOTH.
16. The variable is set if EDG@LOC is not BOTH and EDG@VSS is not 0.

Sample REXX Execs

Here are examples of REXX execs you can create to obtain information about your volumes and data sets. These examples are supplied as members EDGXMP1 and EDGXMP2 in the DFSMSrmm SAMPLIB dataset.

EDGXMP1 VOLCHAIN EXEC

Use EDGXMP1 to list all the volumes in a multivolume set of volumes.

```

/*REXX*****
/*
/* VOLCHAIN EXEC - Given any volume serial number it lists all the
/*                volumes in the multivolume set
/*
/* Variables used from LISTVOLUME command:
/*      edg@vol - Volume serial number
/*      edg@pvl - Volume serial number of previous volume in
/*                multivolume chain.
/*      edg@envl - Volume serial number of next volume in
/*                multivolume chain.
/*
/******
arg volser                                /* Use parameter supplied as the
/* volume serial.

Do while volser = ''                      /* No volume serial so ask for one*/
  Say "Enter Volume Serial:"            /* Issue prompt to TSO user
  Pull volser                            /* Get volume serial from TSO user*/
end

Call LISTVOL volser                      /* Set variable information for
/* requested volume.

If result = 0 then                        /* Are variables OK?
do
  nextvol = edg@envl                    /* Save the next volume pointer
  push edg@vol                          /* Put this volume serial on the
/* stack.

/* Chain through the previous
/* volumes, listing each and
/* putting each volume serial on
/* the stack.

```

Figure 218. VOLCHAIN EXEC Sample REXX Exec (Part 1 of 3)

```

Do while (result = 0) & (strip(edg@pvl) ^= '')
  Call LISTVOL edg@pvl          /* Set variable information for */
                                /* previous volume.                */
  If result = 0 then            /* If previous volume exists then */
    Push edg@vol               /* Put its serial number on the  */
                                /* stack.                        */
  End /* of chaining prevvol pointers */

  edg@nvl = nextvol            /* Start the chain at the next   */
                                /* volume of the volume which was */
                                /* listed first.                  */

                                /* Chain through the next volumes */
                                /* listing each and putting each */
                                /* volume serial on the stack.    */
  Do while (result = 0) & (strip(edg@nvl) ^= '')
    Call LISTVOL edg@nvl        /* Set variable information for */
                                /* previous volume.                */
    If result = 0 then          /* If previous volume exists then */
      Queue edg@vol            /* put its serial number on the  */
                                /* stack.                        */
    End /* of chaining nextvol pointers */

    Do queued()                /* For each volume in the multi- */
      pull volser              /* volume chain, pull the serial */
      say volser               /* off the stack and write it to */
    End /* of volume list */    /* the TSO user.                  */
  end /* of successful list */

exit(0)                        /* return to caller              */

```

Figure 218. VOLCHAIN EXEC Sample REXX Exec (Part 2 of 3)

```

LISTVOL:                                /* LISTVOLUME Procedure: */
                                        /* Input parameter: volume serial */
                                        /* Output: */
                                        /* Result=0: Complete set of */
                                        /* listvolume variables */
                                        /* Result=4: Error message */
                                        /* issued to TSO user */

arg volser
sysauth.edgdate = "EUROPEAN"           /* Tell RMM TSO command to return */
                                        /* output as REXX variables and */
                                        /* dates in EUROPEAN (DD/MM/YYYY) */
                                        /* format. */
save_prompt = prompt("OFF")            /* Turn PROMPTing off. */

                                        /* Get volume information from */
                                        /* DFSMSrmm. */
address "TSO" "RMM LISTVOLUME "volser" ALL"
If rc = 0 then
    lvresult = 0                        /* Indicate Successful LISTVOLUME */
else
    do
        drop sysauth.edgdate           /* An error has occurred. Tell */
                                        /* the RMM TSO command to return */
                                        /* output via messages. */
                                        /* Get error information from */
                                        /* DFSMSrmm. */
        say "LISTVOLUME "volser
        address "TSO" "RMM LISTVOLUME "volser
        lvresult = 4                    /* Indicate Unsuccessful */
                                        /* LISTVOLUME. */
    end
junk = prompt(save_prompt)              /* Restore PROMPT status. */
return lvresult                         /* Return to caller. */

```

Figure 218. VOLCHAIN EXEC Sample REXX Exec (Part 3 of 3)

EDGXMP2 DSNLIST EXEC

Use EDGXMP2 to display volume information.

```

/*REXX*****
/*
/* DSNLIST EXEC - Given any volume serial number it displays all the
/* information held by DFSMSrmm about the data sets on
/* the volume.
/*
/* Variables used from SEARCHDATASET command:
/* edg@dsn.0 - number of data sets on the volume.
/* edg@dsn.x - data set name of each of the data sets on
/* volume (x=1 to edg@dsn.0).
/* edg@vol.x - volume serial number (x=1 to edg@dsn.0)
/* edg@seq.x - data set sequence number (x=1 to edg@dsn.0)
/*
/*****
arg volser /* Use parameter supplied as the
/* volume serial.

Do while volser = '' /* No volume serial so ask for one*/
  Say "Enter Volume Serial:" /* Issue prompt to TSO user
  Pull volser /* Get volume serial from TSO user*/
end

sysauth.edgdate = "EUROPEAN" /* Tell RMM TSO command to return
/* output as REXX variables and
/* dates in EUROPEAN (DD/MM/YYYY)
/* format.
save_prompt = prompt("OFF") /* Turn PROMPTing off.
save_msg = msg("OFF") /* Turn messages off.
/* Get information for data sets
/* on the volume
address "TSO" "RMM SEARCHDATASET D(*) VOLUME("volser") LIMIT(*)"
junk = msg(save_msg) /* Restore previous message status*/

If rc = 0 then
do
  drop sysauth.edgdate /* Tell the RMM TSO command to
/* return output via messages.

/* Display data set listed by the
/* Search command until all are
/* displayed or non-zero return
/* code received.

```

Figure 219. DSNLIST EXEC Sample REXX Exec (Part 1 of 2)

```

Do dataset = 1 to edg@dsn.0 while (rc = 0)
  address "TSO" "RMM LISTDATASET '"edg@dsn.dataset"'
    VOLUME("edg@vol.dataset") SEQ("edg@seq.dataset")"
  say "" /* Write a couple of extra blank */
  say "" /* lines */
end
/* complete with a summary */
say edg@dsn.0 "Data sets on volume "volser" displayed."
end
else
do
  drop sysauth.edgdate /* An error has occurred. Tell */
/* the RMM TSO command to return */
/* output via messages. */
/* Get error information from */
/* DFSMSrmm. */
  say "SEARCHDATASET D(*) VOLUME("volser") LIMIT(*)"
  address "TSO" "RMM SEARCHDATASET D(*) VOLUME("volser") LIMIT(*)"
end
junk = prompt(save_prompt) /* Restore PROMPT status. */
exit(0) /* return to caller */

```

Figure 219. DSNLIST EXEC Sample REXX Exec (Part 2 of 2)

Appendix. DFSMSrmm ISPF Dialog Fast Path Commands

This table lists the fast path commands you can use within the DFSMSrmm ISPF Dialog.

Table 42. DFSMSrmm ISPF Dialog Fast Path Commands

Issue The Fast Path Command	To
ADMIN	Display the Administrator Menu.
COMMANDS	Display the Command Menu.
CONTROL	Display the Control Menu.
CONTROL ACTIONS	Display volume moves and actions.
CONTROL LOCDEF	Display location definitions.
CONTROL MASTER	Display control data set information.
CONTROL MNTMSG	Display mount message definitions.
CONTROL REJECT	Display volumes to be rejected.
CONTROL SECURITY	Display security classification rules.
CONTROL SYSTEM	Display system options and defaults.
CONTROL VLPOOLS	Display volume pool definitions.
DATASET	Display the Data Set Menu.
DATASET ADD	Add a data set.
DATASET CHANGE	Change data set information.
DATASET DELETE	Delete data set information.
DATASET DISPLAY	Request data set information.
DATASET SEARCH	Search for data sets.
LIB	Display the Librarian Menu.
LOCAL	Display a menu your location has tailored.
OPTIONS	Display the Dialog Options Menu.
OPTIONS CONFIRM	Display the current setting for the CONFIRM command.
OPTIONS CONFIRM ON	Confirm any deletes or releases before they are processed.
OPTIONS CONFIRM OFF	Process deletes or releases without confirming.
OPTIONS DATE	Display the current date setting.
OPTIONS DATE American	Set the date format to mm/dd/yyyy.
OPTIONS DATE European	Set the date format to dd/mm/yyyy.
OPTIONS DATE Iso	Set the date format to yyyy/mm/dd.
OPTIONS DATE Julian	Set the date format to yyyy/ddd.
OPTIONS SAVE	Display the current setting for the SAVE command.
OPTIONS SAVE ON	Save commands in a data set for background processing.
OPTIONS SAVE OFF	Process commands interactively.
OPTIONS SORT	Display the Dialog Sort Options Menu.
OPTIONS SORT DATASET	Display the Data Set List Sort Options panel.
OPTIONS SORT PRODUCT	Display the Product List Sort Options panel.
OPTIONS SORT RACK	Display the Rack and Bin List Sort Options panel.

Table 42. DFSMSrmm ISPF Dialog Fast Path Commands (continued)

Issue The Fast Path Command	To
OPTIONS SORT VOLUME	Display the Volume List Sort Options panel.
OPTIONS SORT VRS	Display the VRS List Sort Options panel.
OPTIONS USER	Display the Dialog User Options panel.
OWNER	Display the Owner Menu.
OWNER ADD owner ID	Add a new owner.
OWNER CHANGE owner ID	Change owner information.
OWNER DELETE owner ID	Delete owner information.
OWNER DISPLAY owner ID	Request owner information.
PRODUCT	Display the Product Menu.
PRODUCT ADD	Add a new software product or software product volume.
PRODUCT CHANGE	Change software product information.
PRODUCT DELETE	Delete software product information.
PRODUCT DISPLAY	Request software product information.
PRODUCT SEARCH	Search for software products.
RACK	Display the Rack and Bin Menu.
RACK ADD	Add new rack or bin numbers.
RACK DELETE	Delete rack or bin numbers.
RACK DISPLAY	Request rack or bin number information.
RACK SEARCH	Search for rack or bin numbers.
SUPPORT	Display the Support Menu.
USER	Display the User Menu.
VOLUME	Display the Volume Menu.
VOLUME ADD	Add a new volume.
VOLUME ADDSCR	Add scratch volumes.
VOLUME CHANGE	Change volume information.
VOLUME CONFIRM	Confirm volume moves or actions.
VOLUME DISPLAY	Request volume information.
VOLUME RELEASE	Release any volume.
VOLUME RELUSER	Release owned volumes.
VOLUME REQUEST	Request a volume.
VOLUME SEARCH	Search for volumes.
VRS	Display the Vital Record Specification Menu.
VRS ADD	Add a vital record specification.
VRS CHANGE	Change a vital record specification.
VRS DELETE	Delete a vital record specification.
VRS DISPLAY	Request vital record specification information.
VRS SEARCH	Search for vital record specifications.

Abbreviations

The following abbreviations are defined as they are used in the DFSMSrmm library. If you do not find the abbreviation you are looking for, see *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

This list may include acronyms and abbreviations from:

- *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.
- *Information Technology Vocabulary* developed by subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1).

A

abend. Abnormal end of task

AL. American National Standards Label

| **AMODE.** Addressing mode

ANSI. American National Standards Institute

APAR. Authorized program analysis report

| **API.** Application Programming Interface

ASA. American Standards Association

AUL. ANSI and user header or trailer label

B

BLP. Bypass label processing

| **BTLS.** Basic Tape Library Support

OPC/ESA. Operations Planning and Control/Enterprise Systems Architecture

D

DASD. Direct access storage device

DCB. Data control block

E

EREP. Environmental Record Editing and Printing program

F

FIPS. Federal Information Processing Standard

FMID. Function modification identifier

G

GDG. Generation data group

GDS. Generation data set

GRS. Global resource serialization

I

ID. Identifier

IPL. Initial program load

ISPF. Interactive System Productivity Facility

ISMF. Interactive Storage Management Facility

ISO. International Organization for Standardization

J

JCL. Job control language

JES2. Job entry subsystem 2

JES3. Job entry subsystem 3

JFCB. Job file control block

N

NL. No label

NSL. Nonstandard label

O

OAM. Object Access Method

OPC/ESA. Operations Planning and Control/Enterprise Systems Architecture

P

PF. Program function key

PTF. Program temporary fix

PUT. Program Update Tape

R

RACF. Resource Access Control Facility

REXX. Restructured Extended Executor Language

RMF. Resource Measurement Facility

RMODE. Residence mode

S

SAF. System Authorization Facility

SDB. structured database

SL. Standard label

SMF. System management facility

SMP/E. System Modification Program Extended

SUL. IBM standard and user header or trailer label

SVC. Supervisor call

T

| **TCDB.** Tape Configuration Database

TSO. Time Sharing Option

V

VOLSER. Volume serial number

| **VTs.** Virtual Tape Server

Glossary

The following terms are defined as they are used in the DFSMS/MVS Library. If you do not find the term you are looking for, see the IBM Software Glossary: <http://www.networking.ibm.com/nsg/nsg>

This glossary is an ever-evolving document that defines technical terms used in the documentation for many IBM software products.

A

ANDVRS. An RMM ADDVRS TSO subcommand operand. See Using AND.

assigned date. The date that the volume is assigned to the current owner. Assigned date is not meaningful for a scratch volume.

automated tape library. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. See also *tape library*. Contrast with *manual tape library*.

automatic cartridge loader. An optional feature of the 3480 Magnetic Tape Subsystem that allows preloading of multiple tape cartridges. This feature is standard in the 3490 Magnetic Tape Subsystem.

automatic recording. In DFSMSrmm, the process of recording information about a volume and the data sets on the volume in the DFSMSrmm control data set at open or close time.

availability. For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

B

backup. The process of creating a copy of a data set or object to be used in case of accidental loss.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. See also *integrated catalog facility catalog*.

bin number. The specific shelf location where a volume resides in a storage location; equivalent to a rack number in the removable media library. See also *shelf location*

built-in storage location. One of the Removable Media Manager defined storage locations: LOCAL, DISTANT, and REMOTE.

C

cache fast write. A storage control capability in which the data is written directly to cache without using nonvolatile storage. Cache fast write is useful for temporary data or data that is readily recreated, such as the sort work files created by DFSORT. Contrast with *DASD fast write*.

cartridge eject. For an IBM 3494 Tape Library Dataserver, IBM 3495 Tape Library Dataserver, or an IBM Model M10 3495 Tape Library Dataserver the act of physically removing a tape cartridge usually under robot control, by placing it in an output station. The software logically removes the cartridge by deleting or updating the tape volume record in the tape configuration database. For a manual tape library dataserver, the act of logically removing a tape cartridge from the manual tape library dataserver by deleting or updating the tape volume record in the tape configuration database.

cartridge entry. For either an IBM 3494 Tape Library Dataserver, IBM 3495 Tape Library Dataserver, or a IBM Model M10 3495 Tape Library Dataserver, the process of logically adding a tape cartridge to the library by creating or updating the tape volume record in the tape configuration database. The cartridge entry process includes the assignment of the cartridge to a category — either scratch or private — in the library.

Cartridge System Tape. The base tape cartridge media used with 3480 or 3490 Magnetic Tape Subsystems. Contrast with *Enhanced Capacity Cartridge System Tape*.

cell. A single cartridge location within an automated tape library dataserver. See also *rack number*.

circular file. A type of file that appends data until full. Then, starting at the beginning of the file, subsequent incoming data overwrites the data already there.

command line. On a display screen, a display line usually at the bottom of the screen in which only commands can be entered.

concurrent copy. A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

confirmation panel. A DFSMSrmm panel that lets you tell DFSMSrmm to continue or stop a delete or release action. You specify whether or not you want to confirm delete or release requests in your dialog user options.

container. A receptacle in which one or more exported logical volumes can be stored. A stacked volume containing one or more logical volumes and residing

outside a virtual tape server library is considered to be the container for those volumes.

container volume. See container.

control data set. A VSAM key-sequenced data set that contains the complete inventory of your removable media library, as well as the movement and retention policies you define. In the control data set DFSMSrmm records all changes made to the inventory, such as adding or deleting volumes.

control data set ID. A one-to-eight character identifier for the DFSMSrmm control data set used to ensure that, in a multi-system, multi-complex environment, the correct management functions are performed.

convenience input. The process of adding a small number of tape cartridges to the IBM 3494 Tape Library Dataserver and IBM 3495 Tape Library Dataserver without interrupting operations, by inserting the cartridges directly into cells in a convenience input station.

convenience input/output station. A transfer station with combined tape cartridge input and output functions in the IBM 3494 Tape Library Dataservers only.

convenience input station. A transfer station, used by the operator to add tape cartridges to the IBM 3494 Tape Library Dataserver or an IBM 3495 Tape Library Dataserver, which is accessible from outside the enclosure area.

convenience output. The process of removing a small number of tape cartridges from the IBM 3494 Tape Library Dataserver or an IBM 3495 Tape Library Dataserver without interrupting operations, by removing the cartridges directly from cells in a convenience input station.

convenience output station. A transfer station, used by the operator to remove tape cartridges from the automated tape library dataserver, which is accessible from outside the enclosure area.

conversion. In DFSMSrmm, the process of moving your removable media library inventory from another media management system to DFSMSrmm. DFSMSrmm manages the inventory and policies once you have converted it.

create date. Create date for a dataset is the date that the dataset is written to tape. Create date can also be the date a data set was read if it was created before DFSMSrmm is in use. Create date is updated each time a data set is replaced and not extended. Create date for volumes and other resources defined to DFSMSrmm is the date the resource is defined to DFSMSrmm or the date specified on the command as the create date.

D

DASD fast write. An extended function of some models of the IBM 3990 Storage Control in which data is written concurrently to cache and nonvolatile storage and automatically scheduled for destaging to DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. Use of DASD fast write for system-managed data sets is controlled by storage class attributes to improve performance. See also *dynamic cache management*. Contrast with *cache fast write*.

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also *volume*, *primary storage*, *migration level 1*, *migration level 2*.

data column. A vertical arrangement of identical data items, used on list panels to display an attribute, characteristic, or value of one or more objects.

data entry panel. A panel in which the user communicates with the system by filling in one or more fields.

Data Facility Sort (DFSORT). An IBM licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copying operations, as well as providing versatile data manipulation at the record, field, and bit level.

device. This term is used interchangeably with unit. You mount a tape on a unit or device, such as a 3490.

DFSMS environment. An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the DFSMS environment for MVS, the function is provided by DFSORT, RACF, and the combination of DFSMS/MVS and MVS.

DFSMS/MVS. An IBM System/390 licensed program that provides storage, data, and device management functions. When combined with MVS/ESA SP Version 5 it composes the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMSHsm, and DFSMSrmm.

DFSMSdfp. A DFSMS/MVS functional component or base element of OS/390, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS/MVS functional component or base element of OS/390, used to copy, move, dump, and restore data sets and volumes.

DFSMSHsm. A DFSMS/MVS functional component or base element of OS/390, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMSHsm-managed volume. (1) A primary storage volume, which is defined to DFSMSHsm but which does not belong to a storage group. (2) A volume in a storage group, which is using DFSMSHsm automatic dump, migration, or backup services. Contrast with *system-managed volume* and *DFSMSrmm-managed volume*.

DFSMSHsm-owned volume. A storage volume on which DFSMSHsm stores backup versions, dump copies, or migrated data sets.

DFSMSrmm. A DFSMS/MVS functional component or base element of OS/390, that manages removable media.

DFSMSrmm control data set. See *control data set*.

DFSMSrmm-managed volume. A tape volume that is defined to DFSMSrmm. Contrast with *system-managed volume* and *DFSMSHsm-managed volume*.

disaster recovery. A procedure for copying and storing an installation's essential business data in a secure location, and for recovering that data in the event of a catastrophic problem. Compare with *vital records*.

DISTANT. A DFSMSrmm built-in storage location ID. See *built-in storage location*.

dual copy. A high availability function made possible by nonvolatile storage in some models of the IBM 3990 Storage Control. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical 3990 subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

dump class. A set of characteristics that describes how volume dumps are managed by DFSMSHsm.

duplexing. The process of writing two sets of identical records in order to create a second copy of data.

dynamic cache management. A function that automatically determines which data sets will be cached based on the 3990 subsystem load, the characteristics of the data set, and the performance requirements defined by the storage administrator.

E

eject. The process used to remove a volume from a system-managed library. For an automated tape library dataser, the volume is removed from its cell location and moved to the output station. For a manual tape library dataser, the volume is not moved, but the

tape configuration database is updated to show the volume no longer resides in the manual tape library dataser.

Enhanced Capacity Cartridge System Tape.

Cartridge system tape with increased capacity that can only be used with 3490E Magnetic Tape Subsystems. Contrast with *Cartridge System Tape*.

entry panel. See *data entry panel*.

expiration. The process by which data sets and volumes are identified as available for reuse. In DFSMSrmm, all volumes have an expiration date or retention period set for them either by vital record specification policy, by user-specified JCL when writing a data set to the volume, or by an installation default. When a volume reaches its expiration date or retention period, it becomes eligible for release.

expiration date. The date at which a file is no longer protected against automatic deletion by the system.

expiration processing. The process of inventory management that ensures expired volumes are released and carries out required release actions on those volumes.

export. The operation to remove one or more logical volumes from a virtual tape server library. First, the list of logical volumes to export must be written on an export list volume and then, the export operation itself must be initiated.

export list volume. A virtual tape server logical volume containing the list of logical volumes to export.

exported logical volume. A logical volume that has gone through the export process and now resides on a stacked volume outside a virtual tape server library.

external label. A label attached to the outside of a tape cartridge that is to be stored in an IBM 3494 Tape Library Dataserver or IBM 3495 Tape Library Dataserver. The label might contain the DFSMSrmm rack number of the tape volume.

extract data set. A data set that you use to generate reports.

F

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully or partially-qualified data set names or of certain data set characteristics.

G

generation data group (GDG). A collection of data sets kept in chronological order. Each data set is a generation data set.

generation number. The number of a generation within a generation data group. A zero represents the most current generation of the group, a negative integer (-1) represents an older generation and, a positive integer (+1) represents a new generation that has not yet been cataloged.

guaranteed space. A storage class attribute indicating the space is to be preallocated when a data set is created. If you specify explicit volume serial numbers, SMS honors them. If space to satisfy the allocation is not available on the user-specified volumes, the allocation fails.

H

hardware configuration definition (HCD). An interactive interface in MVS that enables an installation to define hardware configurations from a single point of control.

high capacity input station. A transfer station, used by the operator to add tape cartridges to the IBM 3494 Tape Library Dataserver or IBM 3495 Tape Library Dataserver, which is inside the enclosure area.

high capacity output station. A transfer station, used by the operator to remove tape cartridges from the automated tape library dataserver, which is inside the enclosure area.

home. See *home location*.

home location. For DFSMSrmm, the place where DFSMSrmm normally returns a volume when the volume is no longer retained by vital records processing.

I

ICETOOL. DFSORT's multipurpose data processing and reporting utility.

import. The operation to enter previously exported logical volumes residing on a stacked volume into a virtual tape server library. First, the list of logical volumes to import must be written on an import list volume and the stacked volumes must be entered, and then, the import operation itself must be initiated.

import list volume. A virtual tape server logical volume containing the list of logical volumes to import. This list can contain individual logical volumes to import and/or it can contain a list of stacked volumes in which all logical volumes on the stacked volume are imported.

imported logical volume. An exported logical volume that has gone through the import process and can be referenced as a tape volume within a virtual tape server library. An imported logical volume originates from a stacked volume that went through the export process.

improved data recording capability (IDRC). A recording mode that can increase the effective cartridge data capacity and the effective data rate when enabled and used. IDRC is always enabled on the 3490E Magnetic Tape Subsystem.

installation defined storage location. A storage location defined using the LOCDEF command in the EDGRMMxx parmlib member.

integrated catalog facility catalog. A catalog that is composed of a basic catalog structure (BCS) and its related volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). See also *basic catalog structure* and *VSAM volume data set*.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run interactive, panel-driven dialogs.

in transit. A volume is in transit when it must be moved from one location to another and DFSMSrmm believes that the move has started, but has not yet received confirmation that the move is complete. For a volume moving from a system-managed library, the move starts when the volume is ejected.

internal label. The internal label for standard label tapes is recorded in the VOL1 header label, magnetically recorded on the tape media.

inventory management. The regular tasks that need to be performed to maintain the control data set. See also *expiration processing*, *storage location management processing*, and *vital record processing*.

J

journal. A sequential data set that contains a chronological record of changes made to the DFSMSrmm control data set. You use the journal when you need to reconstruct the DFSMSrmm control data set.

K

keyword. A predefined word that is used as an identifier.

L

Library Control System. The Object Access Method component that controls optical and tape library operations and maintains configuration information.

LOCAL. A DFSMSrmm built-in storage location ID. See *built-in storage location*.

location name. A name given to a place for removable media that DFSMSrmm manages. A location name can be the name of a system-managed library, a storage location name, or the location *SHELF*, identifying shelf space outside a system-managed library or storage locations.

logical volume. A tape volume that resides in a virtual tape server library either in DASD (in the tape volume cache as a virtual volume) or on a stacked volume (as a logical volume) and is referenced from the host as if it were a physical tape volume. From a host perspective, references to a logical volume implies either place of storage in a virtual tape server library.

low-on-scratch management. The process by which DFSMSrmm replenishes scratch volumes in a system-managed library when it detects that there are not enough available scratch volumes.

M

management class. A collection of management attributes, defined by the storage administrator, used to control the release of allocated but unused space: to control the retention, migration, and backup of data sets; to control the retention and backup of aggregate groups; and to control the retention, backup, and class transition of objects. If assigned by ACS routine to system-managed tape volumes, can be used to identify a DFSMSrmm vital record specification.

manual cartridge entry processing. The process by which a volume is added to the tape configuration database when it is added to a manual tape library dataser. DFSMSrmm can initiate this process.

manual mode. An operational mode where DFSMSrmm runs without recording volume usage or validating volumes. The DFSMSrmm TSO commands, ISPF dialog, and inventory management functions are all available in manual mode.

manual tape library. A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which can be mounted on those drives. See also *tape library*. Contrast with *automated tape library*.

master system. The MVS system where the master DFSMSrmm control data set resides.

master volume. A private volume that contains data that is available for write processing based on the DFSMSrmm EDGRMMxx parmlib MASTEROVERWRITE operand.

media format. The type of volume, recording format and techniques used to create the data on the volume.

media library. See *removable media library*.

media management system. A program that helps you manage removable media. DFSMSrmm is a media management system.

media name. An up to 8 character value that describes the shape or type of removable media stored in a storage location. Examples of medianame are: SQUARE, ROUND, CARTRDGE, 3480

media type. A value that specifies the volume's media type. Media type can be specified as: *, CST, or ECCST.

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. The data must be recalled to be used again.

migration level 1. DFSMSHsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 2*.

migration level 2. DFSMSHsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 1*.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

N

name vital record specification. A vital record specification used to define additional retention and movement policy information for data sets or volumes.

NEXTVRS. An RMM ADDVRS TSO subcommand operand. See Using NEXT.

non-scratch volume. A volume that is not scratch, which means it has valid or unexpired data on it. Contrast with *scratch*.

O

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

optical disk. A disk that uses laser technology for data storage and retrieval.

optical volume. Storage space on an optical disk, identified by a volume label. See also *volume*.

option line. See *command line*

owner. In DFSMSrmm, a person or group of persons defined as a DFSMSrmm user owning volumes. An owner is defined to DFSMSrmm through an owner ID.

owner ID. In DFSMSrmm, an identifier for DFSMSrmm users who own volumes.

P

parallel. During conversion, when you install DFSMSrmm concurrently with an existing media management system, it is called running in parallel.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with *temporary data set*.

physical stacked volume. See stacked volume..

physical volume. A physical volume is a volume which can be associated with a physical package which is independently mountable by the host operating system from other physical volumes. Each physical volume has a human and/or machine readable external volume serial number label.

pool. A group of shelf locations in the removable media library whose rack numbers share a common prefix. The shelf locations are logically grouped so that the volumes stored there are easier to find and use.

pooling. The process of arranging shelf locations in the removable media library into logical groups.

pool ID. The identifier for a pool. You define pool IDs in parmlib member EDGRMMxx.

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

primary space allocation. Amount of space requested by a user for a data set when it is created. Contrast with *secondary space allocation*.

primary storage. A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1* and *migration level 2*.

private tape volume. A volume assigned to specific individuals or functions.

primary vital record specification. The first retention and movement policy that DFSMSrmm matches to a data set and volume used for disaster recovery and vital record purposes. See also vital record specification and secondary vital record specification.

protect mode. In protect mode, DFSMSrmm validates all volume requests.

pseudo-generation data group. A collection of data sets, using the same data set name pattern, to be managed like a generation data group. The ~ masking character is used in DFSMSrmm to identify the characters in the pattern that change with each generation.

pull list. A list of scratch volumes to be pulled from the library for use.

R

rack number. A six-character identifier that corresponds to a specific volume's shelf location in the installation's removable media library, and is the identifier used on the external label of the volume to identify it. The rack number identifies the pool and the external volume serial number for a volume residing in an automated tape library dataserer. The rack number identifies the pool, the external volume serial, and shelf location number for a volume not residing in an automated tape library dataserer. The rack number is not written by the tape drive. It exists as an entry in the DFSMSrmm control data set and on the external label of the tape. See also *shelf location*

rack pool. A group of shelves that contains volumes that are generally read-only.

ready to scratch. This describes the condition where a volume is eligible for scratch processing while it resides in a storage location. Since no other release actions are required, the volume can be returned to scratch directly from the storage location.

record-only mode. The operating mode where DFSMSrmm records information about volumes as you use them, but does not validate or reject volumes.

recording format. For a tape volume, the format of the data on the tape; for example, 18 tracks or 36 tracks.

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a journal.

relative start generation. Relative generation zero is the latest generation of a tape; Relative generation -1 is the previous generation of that tape. Relative generation -2 is the generation before the previous one.

REMOTE. A DFSMSrmm built-in storage location ID. See *built-in storage location*.

removable media. See *volume*.

removable media library. The volumes that are available for immediate use, and the shelves where they could reside.

Resource Access Control Facility (RACF). An IBM licensed program that provides for access control by identifying and verifying the users to the system; authorizing access to protected resources; logging the detected unauthorized attempts to enter the system; and logging the detected accesses to protected resources.

Restructured Extended Executor (REXX) Language. A general-purpose, high-level programming language, particularly suitable for EXEC procedures or programs for personal computing.

retention date. Retention date can be the date that a data set or volume is retained by a vital record specification or the date of the inventory management run when the data set or volume is no longer retained by a vital record specification.

retention period. The time for which DFSMSrmm retains a volume or data set before considering it for release. You can retain a data set or volume as part of disaster recovery or vital records management. You set a retention period through a vital record specification that overrides a data set's expiration date.

retention type. The types of retention for which DFSMSrmm retains a volume or data set before considering it for release. The retention types for data sets are BYDAYSCYCLE, CYCLES, DAYS, EXTRADAYS, LASTREFERENCEDAYS, UNTILEXPIRED, and WHILECATALOG. The retention types for volumes are DAYS and CYCLE.

S

scratch. The status of a tape volume that is available for general use, because the data on it is incorrect or is no longer needed. You request a scratch volume when you omit the volume serial number on a request for a tape volume mount.

scratch pool. The collection of tape volumes from which requests for scratch tapes can be satisfied. Contrast with *rack pool*.

scratch processing. The process for returning a volume to scratch status once it is no longer in use and has no outstanding release actions pending.

scratch tape. See *scratch volume*.

scratch volume. A tape volume that contains expired data only. See *scratch*.

secondary space allocation. Amount of additional space requested by the user for a data set when primary space is full. Contrast with *primary space allocation*.

secondary vital record specification. The second retention and movement policy that DFSMSrmm matches to a data set and volume used for disaster recovery and vital records purposes. See also vital record specification and primary vital record specification.

shelf. A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

shelf location. A single space on a shelf for storage of removable media. DFSMSrmm defines a shelf location in the removable media library by a rack number, and a shelf location in a storage location by a bin number. See also *rack number* and *bin number*.

shelf-management. Is the function provided to manage the placement of volumes in individual slots in a location. Shelf-management is provided for the removable media library using rack numbers. For storage locations it is optional as defined by the LOCDEF options in parmlib and uses bin numbers.

shelf-resident volume. A volume that resides in a non-system-managed tape library.

shelf space. See *shelf*.

slot. See *shelf location*.

stacked volume. Physical tape media in a virtual tape server library that is used to store one or more logical volumes. If logical volumes have been exported, stacked volume can also reside outside of a virtual tape server on shelf storage as container volumes used to store the exported logical volumes. The stacked volume itself cannot be mounted by the host operating system.

standard label. An IBM standard tape label.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD volumes and optical volumes treated as a single object storage hierarchy.

storage location. A location physically separate from the removable media library where volumes are stored for disaster recovery, backup, and vital records management.

(storage) location dominance. The priority used by DFSMSrmm to decide where to move a volume within the removable media library during vital record specification processing. It covers all the locations; SHELF, storage locations, and system-managed tape libraries.

storage location management processing. The process of inventory management that assigns a shelf location to volumes that have moved as a result of vital record processing. See also *vital record processing*

stripe. In DFSMS/MVS, the portion of a striped data set that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

subsystem. A special MVS task that provides services and functions to other MVS users. Requests for service are made to the subsystem through a standard MVS facility known as the subsystem interface (SSI). Standard MVS subsystems are the master subsystem and the job entry subsystems JES2 and JES3.

subsystem interface (SSI). The means by which system routines request services of the master subsystem, a job entry subsystem, or other subsystems defined to the subsystem interface.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications. See also *DFSMS environment*.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also *tape library*.

system-managed volume. A DASD, optical, or tape volume that belongs to a storage group. Contrast with *DFSMSshm-managed volume* and *DFSMSrmm-managed volume*.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T

tape configuration database. One or more volume catalogs used to maintain records of system-managed tape libraries and tape volumes.

tape librarian. The person who manages the tape library. This person is a specialized storage administrator.

tape library. A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives. See also *system-managed tape library*.

Tape Library Control System (TLCS). IBM program offering 5785-EAW. DFSMSrmm replaces TLCS.

Tape Library Dataserver. A hardware device that maintains the tape inventory associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, and storage of tapes. An automated or manual tape library that supports system-managed storage of tape volumes. IBM's automated tape library dataservers include the IBM 3494 Tape Library Dataserver and the IBM 3495 Tape Library Dataserver. IBM's manual tape library dataserver is the IBM Model M10 3495 Tape Library Dataserver.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also *storage group*.

tape subsystem. A magnetic tape subsystem consisting of a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume. Storage space on tape, identified by a volume label, which contains data sets or objects and available free space. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with *permanent data set*.

U

Until Expired. Allows the use of vital record specification policies for managing retention in a location as long as the volume expiration date has not been reached.

use attribute. (1) The attribute assigned to a DASD volume that controls when the volume can be used to

allocate new data sets; use attributes are *public*, *private*, and *storage*. (2) For system-managed tape volumes, use attributes are *scratch* and *private*.

user volume. A volume assigned to a user, that can contain any data and can be rewritten as many times as the user wishes until the volume expires.

using AND. A method for linking DFSMSrmm vital record specifications to create chains of vital record specifications. DFSMSrmm applies policies in chains using AND only when all the retention criteria are true.

using NEXT. A method for linking DFSMSrmm vital record specifications to create chains of vital record specifications. DFSMSrmm applies policies in chains using NEXT one vital record at a time.

V

virtual input/output (VIO) storage group. A type of storage group that allocates data sets to paging storage, which simulates a DASD volume. VIO storage groups do not contain any actual DASD volumes. See also *storage group*.

Virtual Tape Server (VTS). This subsystem, integrated into the Magstar 3494 Tape Library, combines the random access and high performance characteristics of DASD with outboard hierarchical storage management and virtual tape devices and tape volumes.

vital record group. A set of data sets with the same name that matches to the same DFSMSrmm vital record specification

vital record processing. The process of inventory management that determines which data sets and volumes DFSMSrmm should retain and whether a volume needs to move. These volumes and data sets have been assigned a vital record specification.

vital record specification. Policies defined to manage the retention and movement of data sets and volumes used for disaster recovery and vital records purposes.

vital record specification management value. A one-to-eight character name defined by your installation and used to assign management and retention values to tape data sets. The vital record management value can be any value you chose to create a match between a vital record specification and data sets and volumes in your installation. By matching the vital record specifications to the data set or volumes, DFSMSrmm applies the retention and movement policies you define in the vital record specifications. During inventory management VRSEL processing, DFSMSrmm selects the correct, best matching vital record specification for a tape data set or volume.

vital records. A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement. Compare with *disaster recovery*.

volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *logical volume*, *optical volume*, *stacked volume*, and *tape volume*.

volume catalog. See *tape configuration database*.

volume expiration date. The date the volume should expire based on the highest expiration date of the data sets that reside on the volume.

volume serial number (VOLSER). An identification number in a volume label that is assigned when a volume is prepared for use on the system. For standard label volumes, the volume serial number is the VOL1 label of the volume. For no label volumes, the volume serial number is the name the user assigns to the volume. In DFSMSrmm, volume serial numbers do not have to match rack numbers.

VSAM volume data set (VVDS). A data set that describes the characteristics of VSAM and system-managed data sets residing on a given DASD volume; part of an integrated catalog facility catalog. See also *basic catalog structure* and *integrated catalog facility catalog*.

W

warning mode. The operating mode in which DFSMSrmm validates volumes as you use them, but issues warning messages when it discovers errors instead of rejecting volumes.

Index

Special Characters

% (percent sign) 82

* (asterisk) 82

A

ABEND

reserved data set name mask 83

resetting the flag 274

retention of data set closed by abend processing 6

searching for data sets marked open 363, 380, 381

volume information

non-scratch volume 57, 241

scratch volume 59, 241

ABEND vital record specification 102

accounting information 334

Action bar, how used 46

ADDBIN Subcommand

overview 224

return codes 226

syntax 225

ADDDATASET Subcommand

overview 227

return codes 232

syntax 228

adding

a creating job name 229

an exported logical volume 246

bin numbers 54, 224

bin numbers to a built-in storage location 238

bin numbers to an installation defined location 238

compaction information 245, 292

data set information 73, 227

exported logical volumes 246

ISO/ANSI label for VOL1 251

ISO/ANSI label version 246, 247

logical volumes 253

mediatype information 248, 299

owner ID 63, 233

owner information 63, 233

physical volumes 246, 253

rack number 54, 238

recording formation information 250, 302

scratch volumes 59

shelf location information 224, 238

software product information 66, 236

vital record specification 90, 255

volumes 241

ADDOWNER Subcommand

overview 233

return codes 235

syntax 234

ADDPRODUCT Subcommand

overview 236

return codes 237

syntax 236

ADDRACK Subcommand

overview 238

ADDRACK Subcommand (*continued*)

return codes 238

syntax 239

ADDVOLUME Subcommand

overview 241

return codes 254

syntax 242

ADDVRS example 138

ADDVRS Subcommand

overview 255

return codes 269

syntax 257

ADMIN fast path command 433

ADMIN operand 40

American date format 42

application programming interface

description 14

applying movement and retention policies 83

archiving data 116

assigning

a new volume serial number 299

access to volumes 244

bin numbers 9

priority to table field names 25

scratch volume 28, 326

authorizing users

to use the DFSMSrmm EDGUX100 installation

exit 60

users by defining resources to RACF 16

using STGADMIN.EDG.IGNORE.TAPE 60

using STGADMIN.EDG.LISTCONTROL 171

using STGADMIN.EDG.MASTER 60

automated tape library dataserver

cartridge entry processing 120

confirming moves 124

convenience output station 121

description 2

ejecting volumes from 121, 320

high capacity station 121

automatic

data set information recording 72

deletion of data set information 76

owner notification 64, 153

processing 149

recovery 207

release action 150, 152

releasing of volumes 148

return to scratch status 150

volume information recording 56

volume release 148

AUTOMOVE operand 285

B

background processing 43

batch job

requesting lists and resource details 27

submitting 51, 222

- batch processing 210
- bin number
 - adding 54, 224
 - defining 53, 224
 - deleting 55, 308
 - details 163, 331
 - initial 55, 56
 - line operators 182, 184
 - list 172, 358
- building an export/import list 389
- built-in storage locations
 - adding bin numbers to 224
 - deleting bin numbers from 308
 - description 3
 - displaying bin number details 331
 - priority of moves to 3
- bypass label processing
 - conditions when not permitted 203
 - defined in installation options 4
 - processing support 13

C

- calculating
 - volume retention date 108, 109
- canceling outstanding volume moves 285, 299
- cartridge entry processing 120
- catalog control
 - to retain data sets 100
- catalog date format 109
- catalog retention period 101, 267
- chaining
 - data sets in a multi-volume data set 381
 - data sets in the same multi-volume data set 363
- chaining vital record specifications 7, 79
- CHANGEDATASET Subcommand
 - overview 270
 - return codes 276
 - syntax 271
- CHANGEOWNER Subcommand
 - overview 277
 - return codes 279
 - syntax 277
- CHANGEPRODUCT Subcommand
 - overview 280
 - return codes 281
 - syntax 280
- CHANGEVOLUME Subcommand
 - overview 282
 - return codes 307
 - syntax 286
- changing
 - data set information 270
 - data set information automatically 35
 - data set information with RMM CHANGEDATASET subcommand 73
 - data set name 274
 - dialog processing options 41
 - expiration date 34, 154
 - home location with RMM CHANGEVOLUME subcommand 284

- changing (*continued*)
 - label version 270
 - list sort order 176
 - location name 298
 - owner information 31, 64
 - owner information with RMM CHANGEOWNER subcommand 277
 - pending status or scratch status 283
 - pool ID 122
 - release date 34, 154
 - retention period 34
 - shelf location 122
 - software product information 69, 280
 - vital record specification 94
 - volume information 60, 282
 - volume release date 34, 154
 - volume serial number 299
 - volume type 305
- clearing loan location 297
- clearing the loan location 123
- clearing volume location information 295
- CLIST
 - confirming volume moves 127
 - confirming volume release actions 158
 - creating a 191
 - ejecting volumes 121
- CLIST operand 15
- closed-cycle GDGs 112
- command
 - abbreviations 221
 - ADDBIN, TSO subcommand 224
 - ADDDATASET, TSO subcommand 227
 - ADDOWNER, TSO subcommand 233
 - ADDPRODUCT, TSO subcommand 236
 - ADDRACK, TSO subcommand 238
 - ADDVOLUME, TSO subcommand 241
 - ADDVRS, TSO subcommand 255
 - batch job 222
 - CHANGEDATASET, TSO subcommand 270
 - CHANGEOWNER, TSO subcommand 277
 - CHANGEPRODUCT, TSO subcommand 280
 - CHANGEVOLUME, TSO subcommand 282
 - CONFIRM 43
 - DATE 42
 - DELETEBIN, TSO subcommand 308
 - DELETEDATASET, TSO subcommand 311
 - DELETEOWNER, TSO subcommand 313
 - DELETEPRODUCT, TSO subcommand 315
 - DELETERACK, TSO subcommand 317
 - DELETEVOLUME, TSO subcommand 320
 - DELETEVRS, TSO subcommand 324
 - DFSMSrmm 221
 - END 51
 - fast path 46, 49
 - GETVOLUME, TSO subcommand 326
 - LISTBIN, TSO subcommand 331
 - LISTCONTROL, TSO subcommand 333
 - LISTDATASET, TSO subcommand 342
 - LISTOWNER, TSO subcommand 345
 - LISTPRODUCT, TSO subcommand 347
 - LISTRACK, TSO subcommand 349

- command (*continued*)
 - LISTVOLUME, TSO subcommand 221
 - LISTVRS, TSO subcommand 355
 - PRINT, ISPF command 26, 191
 - PRINT-HI, ISPF command 26, 191
 - processing 43
 - RETURN 51
 - SAVE 43
 - SEARCHBIN, TSO subcommand 358
 - SEARCHDATASET, TSO subcommand 362
 - SEARCHPRODUCT, TSO subcommand 369
 - SEARCHRACK, TSO subcommand 372
 - SEARCHVOLUME, TSO subcommand 376
 - SEARCHVRS, TSO subcommand 392
 - SORT 46, 179
 - specifying 221
 - TSO subcommand 221
 - Command Menu 48
 - COMMAND operand 40
 - COMMANDS fast path command 433
 - compaction information
 - adding 245, 292
 - CONFIRM command 43
 - confirm delete option 43
 - confirming
 - all pending moves 99, 284
 - delete and release requests 43
 - deletions
 - data set information 76
 - software product 71
 - line operators 157
 - list 186
 - manual release action 151, 155
 - using CHANGEVOLUME 124, 284
 - volume move 124, 284
 - volume release 159
 - console messages 194
 - container
 - defining 246
 - CONTROL
 - fast path command 433
 - operand 40
 - control data set
 - backing up 8
 - control record information 333
 - defining resources 53
 - displaying details 333
 - recovering 8
 - system options information 171
 - controlling data set recording 57
 - converting from closed-cycle gdgs to scratch pools 112
 - correcting data set name 274
 - correcting volume serial number 299
 - creating
 - a CLIST 191
 - bin number list 172
 - creating a volume export list 143
 - creating a volume import list 145
 - data set list 172
 - drop/ship list 186
 - list of resources 172
 - creating (*continued*)
 - rack number list 191
 - reports 14
 - REXX exec 411
 - shelf location list 172
 - software product list 172
 - vital record specifications list 172
 - volume list 172
 - volume movements and actions list 190
 - creating job name
 - changing with RMM CHANGEDATASET
 - subcommand 276
 - default used by DFSMSrmm 273
 - definition 102
 - displaying with RMM LISTDATASET
 - subcommand 342
 - specifying with RMM ADDDATASET
 - subcommand 229
 - cycles date format 109
- ## D
- data class recorded by DFSMSrmm 72
 - data set
 - adding information 73, 227
 - automatic recording of information 72
 - chaining data sets in a multi-volume data set 381
 - chaining data sets in the same multi-volume data set 363
 - changing information 73, 270
 - changing information automatically 35
 - changing name 274
 - defining 72, 227
 - deleting information 76, 311
 - details 163, 342
 - fully qualified name 82
 - generic data set name 82
 - how to define data set name 82
 - line operators 182, 184
 - list 172, 362
 - management class 342
 - managing 13
 - name mask 263
 - non-restricted operands or fields 270
 - restricted operands or fields 270
 - retain by job name 83
 - retaining based on catalog status 100
 - retaining by job name 102
 - retaining until expired 100, 101
 - retaining while cataloged 100
 - retaining while open 101
 - retention date 108
 - saved commands 44, 51
 - searching for data sets in a multi-volume data set 363, 381
 - searching for data sets marked open 363, 380
 - sequence number 74, 164
 - types of retention 5
 - uncataloging 77
 - updating information 73, 270
 - updating information automatically 35
 - vital record specification management value 342

- data set recording 57, 351
- data set vital record specification
 - adding 94
 - definition 79
 - example 7
 - matching data set names 84, 85
 - specifying data set names 81
- DATASET
 - fast path command 433
 - operand 40
- DATE command 42
- date format
 - catalog date 109
 - cycles date 109
 - setting for a session 42
 - used in initialization messages 197
- defining
 - a creating job name 229
 - a scratch volume 57
 - data set 72
 - exported logical volumes 246
 - home location 5, 113
 - library names 114
 - loan location 123
 - logical volumes 253
 - master volume 57
 - movement policies 113
 - original expiration date 154
 - owner 63, 233
 - physical volumes 253
 - resource 53
 - retention and movement policies 5
 - retention policies 99
 - scratch volume 59
 - shelf location 53, 224, 238
 - software product 66, 236
 - user volume 57
 - vital record specification 90
 - volume 56, 241
- DELETEBIN Subcommand
 - overview 308
 - return codes 310
 - syntax 309
- DELETEDATASET Subcommand
 - overview 311
 - return codes 312
 - syntax 311
- DELETEOWNER Subcommand
 - overview 313
 - return codes 313
 - syntax 313
- DELETEPRODUCT Subcommand
 - overview 315
 - return codes 315
 - syntax 315
- DELETERACK Subcommand
 - overview 317
 - return codes 319
 - syntax 318
- DELETEVOLUME Subcommand
 - overview 320

- DELETEVOLUME Subcommand *(continued)*
 - return codes 320
 - syntax 321
- DELETEVRS Subcommand
 - overview 324
 - return codes 325
 - syntax 324
- deleting
 - an empty rack with a medianame that does not match VLPOOL 319
 - bin number 55, 308
 - bin numbers from a built-in storage location 317
 - bin numbers from an installation defined location 317
 - confirm
 - data set information 76
 - software product information 71
 - data set information 76, 311
 - owner ID 65, 313
 - owner information 65, 313
 - rack number 55, 317
 - shelf location information 55, 317
 - software product information 71, 315
 - vital record specification 96, 324
 - volume information 62, 320
- device types supported 2, 3
- DFSMSdftp labeling support 247, 297
- DFSMSHsm
 - backup and migration tapes 256
 - migration tapes, moving 116
 - retaining DFSMSHsm ABARS backups 268
 - validating data set name 17
- DFSMSrmm
 - description 1
 - general user tasks 20
 - ISPF dialog 23, 39
 - modes of operation 17
 - navigating through the DFSMSrmm dialog 46
 - operator tasks 21
 - quick reference 23
 - reason codes 401
 - refreshing the DFSMSrmm installation exits 194
 - removing DFSMSrmm from the system 195
 - resources 53
 - return codes 401
 - running utilities 8
 - starting 23, 39
 - stopping 24, 50
 - storage administrator tasks 20
 - system programmer tasks 20
 - tape librarian tasks 20
 - tape mount validation rules 17
 - TSO subcommands 221
 - validating tape volumes 17
 - volume rejection rules 18
 - wrong label processing 211
- DFSMSrmm ISPF exec
 - operands 40
 - using 40
- DFSORT 16
- diagnosing errors 40, 150

- dialog
 - accessing functions 46
 - exiting 24, 50
 - fast path commands 49
 - help panels 50
 - menu types 47
 - options 41
 - starting 23, 39
- disaster recovery 99
- displaying
 - bin number details 331, 349
 - control record information 333
 - data set details 342
 - DFSMSrmm control data set information 171
 - disposition control options 171
 - installation options 171
 - owner details 345
 - parmlib options information 333
 - rack number details 349
 - retention date 351
 - shelf location details 331, 349
 - software product details 347
 - vital record specification details 355
 - volume details 351
- disposition control
 - default sticky label description 218
 - displaying options 171
 - sticky label support 217
 - system options information 334
- DISTANT storage location 3
- drop/ship list 186
- DSNLIST EXEC 429
- dummy tag
 - ABEND 102
 - matching to data set names 84, 85
 - OPEN 101
 - retaining 99
- duplicate volume serial number 60
 - ignoring with EDGUX100 60
 - managing 68
 - redefining 60

E

- EDG019VM installation exit 202
- EDGINERS utility
 - automatic processing 152
 - differences and similarities to IEHINITT 8
 - manual processing 208
 - using EDGINERS 209
 - wrong label processing 211
- EDGJIMPC sample 145
- EDGLABEL procedure
 - description 208
 - message responses 212
- EDGRPTD utility 163
- EDGUX100 installation exit 60
- EDGXMP1 VOLCHAIN EXEC 426
- EDGXMP2 DSNLIST EXEC 429

- ejecting
 - volumes by building a list of DFSMSrmm subcommands 121
 - volumes from tape libraries 120
 - volumes with ISMF or MVS console command 306
 - volumes with RMM DELETEVOLUME subcommand 320
- empty rack or bin number 55
- END command 51
- ending a search for volume information 383
- ERASE request format 215
- erase volume release action 34, 152
- erasing volumes 152, 208
- error diagnosis 40, 150
- European date format 42
- examples
 - adding
 - a data set with RMM ADDVOLUME subcommand 227
 - a new owner with RMM ADDOWNER subcommand 233
 - a new product with RMM ADDPRODUCT subcommand 236
 - empty bin to the storage location with RMM ADDBIN subcommand 224
 - empty bins to a storage location with RMM ADDBIN subcommand 226
 - empty bins to a storage location with RMM ADDRACK subcommand 240
 - empty racks to the removable media library with RMM ADDRACK subcommand 238
 - new scratch volumes with RMM ADDVOLUME subcommand 241
 - tape cartridge with RMM ADDVOLUME subcommand 254
 - volumes in a system-managed library with RMM ADDVOLUME subcommand 254
 - volumes using status information from the TCDB 254
 - associating
 - a volume to a software product with RMM CHANGEVOLUME subcommand 306
 - building
 - a list of eject commands with RMM SEARCHVOLUME subcommand 388
 - changing
 - owner information with RMM CHANGEOWNER subcommand 277
 - owner of a software product with RMM CHANGEPRODUCT subcommand 280
 - the MANAGEMENTVALUE assigned to a data set 276
 - confirming
 - volume actions with RMM CHANGEVOLUME subcommand 306
 - confirming volume moves for exported volumes 144
 - creating
 - a list of data sets with RMM SEARCHDATASET subcommand 362
 - a list of empty or available bin numbers with RMM SEARCHBIN subcommand 358

examples (*continued*)

- a list of scratch volumes with RMM
SEARCHRACK subcommand 227
- a list of software products with RMM
SEARCHPRODUCT subcommand 369
- a list of volumes that belong to an owner with
RMM SEARCHVOLUME subcommand 377
- creating a volume export list 143
- creating a volume import list 145
- defining
 - vital record specification with RMM ADDVRS
subcommand 267
- deleting
 - a data set vital record specification 325
 - a volume vital record specification with RMM
DELETEVRS subcommand 324
 - information about a volume with RMM
DELETEVOLUME subcommand 321
 - information about data sets with RMM
DELETEDATASET subcommand 311
 - software product information with RMM
DELETEPRODUCT subcommand 315
 - user ID and node for an owner with RMM
CHANGEOWNER subcommand 278
 - user IDs with RMM CHANGEVOLUME
subcommand 285
- displaying
 - information for a single data set with RMM
LISTDATASET subcommand 342
 - installation's options and rules with RMM
LISTCONTROL subcommand 333
 - location definitions with RMM LISTCONTROL
subcommand 340
 - security class masks with RMM LISTCONTROL
subcommand 339
- extending
 - volume retention period with RMM
CHANGEVOLUME subcommand 306
- JCL to run a batch TSO TMP 222
- listing
 - vital record specifications with RMM SEARCHVRS
subcommand 393
- managing
 - special dates with RMM ADDVRS
subcommand 268
- marking
 - a volume as no longer in transit with RMM
CHANGEVOLUME subcommand 306
- moving
 - a volume to an automated tape library dataserer
with RMM CHANGEVOLUME
subcommand 306
- releasing
 - volumes associated with a software product with
RMM DELETEPRODUCT subcommand 315
- removing
 - bin numbers with RMM DELETEDBIN
subcommand 310
 - bin numbers with RMM DELETERACK
subcommand 319

examples (*continued*)

- removing (*continued*)
 - shelf location information with RMM DELETEDBIN
subcommand 310
 - shelf location information with RMM
DELETERACK subcommand 317
 - requesting
 - a volume from the default scratch pool with RMM
GETVOLUME subcommand 326
 - information a shelf location with RMM LISTBIN
subcommand 331
 - information a shelf location with RMM LISTRACK
subcommand 349
 - information about a vital record specification with
RMM LISTVRS subcommand 355
 - owner information with RMM LISTOWNER
subcommand 345
 - software product information with RMM
LISTPRODUCT subcommand 347
 - volume serial information with RMM
LISTVOLUME subcommand 351
 - volume statistics with RMM LISTVOLUME
subcommand 353
 - retaining
 - DFSMSShsm ABARS backups with RMM ADDVRS
subcommand 268
 - DFSMSShsm backup and migration tapes with
RMM ADDVRS subcommand 256
 - retention by cycles 81
 - setting
 - a creating job name 276
 - testing
 - return codes in a CLIST 401
 - return codes in a REXX procedure 401
 - transferring
 - volume ownership with RMM DELETEDOWNER
subcommand 313
 - updating
 - data set information with RMM
CHANGEDATASET subcommand 270
 - using TSO help 222
- exec
- DFSMSrmm ISPF 40
 - REXX 411
- Exit Menu 51
- exiting the DFSMSrmm dialog 24, 50
- expiration
- date 34, 154
 - JCL date 103
 - processing 148
- expiration date protection 103, 154
- export processing 143
- exported logical volume
- adding 246
- EXTRADAYS example 138

F

- fast path commands
 - issuing 49
 - using in DFSMSrmm ISPF dialog panels 46
- fields, restricted and non-restricted 270, 282

- filtering with data set name masks 83
- foreground processing 43
- formatting a list of logical volumes from RMM
 - SEARCHVOLUME CLIST output 145
- forward recovery 8
- freeing up space in system-managed libraries 114
- fully qualified data set name 82
- function menus 48

G

- general user
 - quick reference for 23
 - tasks 20
- generation data group (GDG)
 - base names 88
 - closed-cycle 112
 - converting to use scratch 113
- generic data set
 - description 82
 - using 263
- generic data set name 82
- generic job name description 85
- generic volume serial number 7
- GETVOLUME Subcommand
 - overview 326
 - return codes 330
 - syntax 327
- global confirmation
 - examples 126
 - performing 284
 - undoing movement and action confirmation 284
- grouping volumes 38

H

- help
 - online RMM TSO subcommand help 222
 - panels 50
- HELP command 50
- help facility 50
- hierarchy of moves 3
- home location
 - changing 114
 - changing with CHANGEVOLUME
 - subcommand 284
 - definition 5
 - setting 113, 248

I

- I/O errors on a volume 11, 150
- IBM standard and user header or trailer labels (SUL) 12
- IBM standard labels (SL) 12
- ICETOOL, DFSORT utility
 - description 14, 16
- IEHINITT
 - description 8
- IFG019VM installation exit 202
- ignoring duplicate volume serial numbers 60

- implementing a data archive process 116
- import processing 144
- in transit 124
- information about your resources 24
- information recording performed by DFSMSrmm 72
- INIT request format 215
- initial
 - rack or bin number 55, 56
 - volume serial number 59
- initialization messages 197
- initialize volume release action 34, 152
- initializing volumes 152, 208
- installation defined storage locations
 - adding bin numbers to 224
 - deleting bin numbers from 308
 - description 3
 - displaying bin number details 331
 - obtaining information about 55
 - priority of moves to 3
- installation exits
 - EDGUX100 60
 - messages 207
 - refreshing the DFSMSrmm installation exits 194
- inventory management
 - confirming volume movements 124
 - description 8
 - inventory management 8
 - report 163
 - trial run vital record processing 8
- ISO/ANSI
 - changing label version 295
 - changing the label version 304
 - defining label version 246, 247
 - defining label version in the VOL1 label 251
 - labeling 209
 - searching for ISO/ANSI label information 382
 - usage 12
 - Version 3 210
 - Version 4 210
- ISO/ANSI and user header or trailer labels (AUL) 12
- ISO/ANSI labels (AL) 12
- ISO date format 42
- ISPF dialog 39

J

- JCL expiration date 103
- job name
 - controlling policy selection 83
 - creating job name definition 102
 - description 5
 - generic job name 85
 - how to define 85
 - retaining data sets by job name 102
 - using with ADDDATASET subcommand 227
- job name mask 85
- journal
 - operator system error messages 206
 - rexx variable 419
 - system options information 171, 334
- Julian date format 42

L

- LABEL procedure
 - description 208
- label types supported 12
- labeling ISO/ANSI tape labels 209
- last referenced retention type 100
- LIBRARIAN fast path command 433
- LIBRARIAN operand 40
- library
 - automated tape 1
 - manual tape 1
 - non-system-managed 1
 - removable media 1
 - system-managed 1
- limit, search 172
- limiting the search for volume information 383
- line operators
 - confirm 157, 191
 - data set 182
 - rack and bin number 182
 - release volumes 161
 - shelf location 182
 - software product 182
 - using 182
 - vital record specification 182
 - volume 182
- LISTBIN Subcommand
 - overview 331
 - return codes 332
 - syntax 331
- LISTCONTROL Subcommand
 - overview 333
 - return codes 341
 - syntax 335
- LISTDATASET Subcommand
 - overview 342
 - return codes 344
 - syntax 343
- LISTOWNER Subcommand
 - overview 345
 - return codes 345
 - syntax 345
- LISTPRODUCT Subcommand
 - overview 347
 - return codes 348
 - syntax 347
- LISTTRACK Subcommand
 - overview 349
 - return codes 350
 - syntax 349
- lists
 - bin number 358, 372
 - changing sort order 176
 - CLISTS of executable commands 191
 - confirm volume moves 186
 - creating 172
 - data set 172, 362
 - drop/ship 186
 - printing 26, 191
 - rack number 372
 - requesting 172

- lists (*continued*)
 - scratch volume 358
 - search limit 172
 - shelf location 358, 372
 - software product 172, 369
 - sorting 25, 44, 176
 - using 172
 - vital record specification 392
 - volume 172, 376
- LISTVOLUME Subcommand
 - overview 351
 - return codes 354
 - syntax 352
- LISTVRS Subcommand
 - overview 355
 - return codes 357
 - syntax 356
- loan location
 - changing 305
 - clearing loan location 297
 - defining 246
 - moving to a 122
 - returning from a 123
- LOCAL fast path command 433
- LOCAL operand 40
- LOCAL storage location 3
- location
 - changing 298
 - specifying move locations 7
- location definitions 340
- LOCDEF command 116
- logical volume, changing 305

M

- management class
 - defining management and retention policy 255
 - displaying data set information 342
 - recorded by DFSMSrmm 72
 - specifying 104, 262
- management class names 7
- managing
 - data sets left open by a system failure 101
 - data sets that were closed by ABEND
 - processing 102
 - multiple data sets with data set masks 83
 - special dates 268
 - vital records 99
 - volumes with the same volume serial number 60
- manual move control 117
- manual processing 149
- manual tape library dataserver
 - description 2
 - moving volumes to 120
- MANUALMOVE operand 285
- mask, data set name 263
- masking characters 88
- master status 9
- master volume
 - definition 57
 - overwriting 57
 - searching for information about 366

- matching order for vital record specifications 84, 85
- maximum retention period 103
- mediatype information
 - adding 248, 299
- menu
 - Command 48
 - Exit 51
 - primary option menu 39
 - user 47
 - using 47
- merging UNTILEXPIRED and WHILECATALOG 105
- messages
 - displaying case 334
 - initialization 197
 - installation exit 207
 - operator 196
 - system error 206
 - tape mount 200
 - tape processing 205
 - tape validation 202
- minimum retention with WHILECATALOG 101, 267
- mode of operation
 - definitions 17
 - description 17
- mount message processing 200
- mouse button 47
- movement hierarchy 7
- movement policies 113
- movement priority number 7
- moving
 - DFSMSHsm migration tapes 116
 - from storage locations 123
 - list 190
 - out of system-managed libraries 114
 - to loan locations 122
 - volumes 99, 122
 - volumes using manual move control 117
 - volumes using vital record specification chains 7
- multivolume chain
 - last volume in a multivolume chain 306

N

- name, data set 263
- name filtering rules 83
- name vital record specification 7
 - adding 94, 256
 - definition 79
 - example 81
- navigating through DFSMSrmm
 - using fast path commands 49
 - using the DFSMSrmm panels 46
- NEXTVRS example 137
- no label
 - DFSMSrmm support 215
 - obtaining information about 202
 - usage 12
- non-restricted operands or fields 270, 282
- non-restricted volume information 282
- non-system-managed tape library 3
- nonstandard tape label 13

- notification of volume release 37
- notify owner release action 34, 153
- notifying owners 64

O

- obtaining
 - information about NL tapes 202
 - information about your resources 24
 - parmlib suffix 334, 413
- online help 222
- OPEN
 - resetting the flag 292
 - searching for data sets marked open 385
- OPEN vital record specification 101
- operands, restricted and non-restricted 270, 282
- operator
 - initialization message responses 197
 - installation exit message responses 207
 - LABEL procedure message responses 212
 - messages 196
 - procedures 193
 - sticky label support 217
 - system error message responses 206
 - tape mount message responses 200
 - tape processing message responses 205
 - tape validation message responses 202
 - tasks 21
- options
 - confirm delete 43
 - date format 42
 - dialog 41
 - list sorting 44
 - sorting 44
- OPTIONS fast path command 433
- overriding Automatic Move Processing 285
- overriding expiration date 38, 103
- owner
 - adding information 63, 233
 - changing information 31, 64, 277
 - defining 63, 233
 - deleting information 65, 313
 - details 163, 345
 - ID
 - adding 63, 233
 - changing 31, 64, 277
 - defining 63, 233
 - deleting 65, 313
 - details 163, 345
 - managing information 14
 - notification 64, 153
 - requesting volume return 151
 - updating information 31, 64, 277
- OWNER fast path command 433
- OWNER operand 40

P

- panel
 - help 50
- parmlib member EDGRMMxx
 - description 5

- parmlib member EDGRMMxx *(continued)*
 - suffix 5, 413
- parmlib options
 - displaying details 333
 - implementing new options 194
- pending movement and action
 - confirming 99, 284
 - confirming using the DFSMSrmm ISPF Dialog 125
 - for return to scratch volumes 163
- permanent I/O errors 150
- physical volume
 - changing 305
- policy
 - defining for data sets and volumes 5
 - defining retention and movement policies 5
 - movement 99
 - retention 99
 - setting up 38
- pool
 - adding rack number 53, 55
 - changing information for a volume 122
 - definition 9
 - details 188
 - moving volumes 122
 - rack 9
 - scratch 9
 - types 9
 - viewing pool ids 58
 - within system-managed libraries 10
- pooling requirements 38
- primary option menu 39
- PRINT command 26, 191
- PRINT-HI command 26, 191
- printing lists and details panels 26, 191
- priority
 - volume moves 3
- private volume
 - definition 9
 - overwriting 57
 - searching for information about 366
- problem determination aid
 - controlling tracing 196
 - initialization messages 200
 - system error messages 207
- problem diagnosis 40
- procedures, operator 193
- processing
 - background 43
 - batch, commands in 51
 - foreground 43
 - save options 44
 - saved commands data set 51
 - saved commands data set, the 44
- processing NL tapes 202
- PRODUCT fast path command 433
- PRODUCT operand 40
- pseudo-GDG
 - definition 88
 - specifying 263
- pulling scratch volumes 185

Q

- quick reference 23
- quiescing DFSMSrmm 195

R

- RACF
 - authorizing users 16
 - STGADMIN.EDG.IGNORE.TAPE 60
 - STGADMIN.EDG.LISTCONTROL 171, 333
 - STGADMIN.EDG.MASTER 60
 - STGADMIN.EDG.RELEASE 320
 - using tape profiles 12
- RACK fast path command 433
- rack number
 - adding 54, 238
 - adding to a pool 55
 - defining 53, 238
 - deleting 55, 317
 - deleting an empty rack with a medianame that does not match VLPOOL 319
 - details 163, 349
 - initial 55, 56
 - line operators 182, 184
 - list 172, 372
 - removing 301
- RACK operand 40
- rack pool
 - definition 9
 - setting up 53
 - viewing information about 55
- ready to scratch
 - movement report 163
 - NOTREADYTOSCRATCH operand 293
 - READYTOSCRATCH operand 293
 - using to confirm volume movement 127, 285
- reason codes 402
- reclaiming volumes from pending status or scratch status 283
- recording
 - data set information 35, 72
 - volume information 31
- recording format information
 - adding 250, 302
- recovery
 - description 8
- redefining duplicate volume serial numbers 60
- refreshing DFSMSrmm 194
- refreshing the DFSMSrmm installation exits 194
- rejecting
 - volumes 18
- relabeling a volume 152
- release action
 - confirming 151, 155
 - list 190
 - setting 148
 - setting a 34
 - types 11
- release date 34
- release notification 37
- release processing 149

- releasing a volume
 - associated with a software product 71
 - confirming 159
 - DELETEVOLUME subcommand 320
 - description 148
 - line operators 161
 - manually 29, 158
 - notifying owner 64, 153
 - types of 34
- REMOTE storage location 3
- removable media library
 - definition 1
 - moving volumes 99
 - pools 53
 - retaining volumes 99
 - shelf locations 53
 - shelves 53
- removing
 - a rack number 301
 - DFSMSrmm from the system 195
 - last volume in a multivolume chain 306
 - shelf location information with RMM DELETEDBIN subcommand 308
 - shelf location information with RMM DELETERRACK subcommand 317
- replace volume release action 34, 150
- replacing
 - volumes with duplicate volume serial numbers 68
 - volumes with permanent errors 150
- report
 - creating 14
 - EDGAUD DFSMSrmm security and audit report 8
 - EDGRPTD DFSMSrmm movement and inventory report 15
 - EDGRRPTE exec 8
 - extract data set 163
 - inventory 163
- requesting
 - bin number details 163, 331
 - bin number list 172, 358
 - confirm delete option 43
 - control record information 333
 - data set details 163, 342
 - data set list 172, 362
 - drop/ship list 186
 - information 163
 - list of resources 172
 - lists 163
 - owner details 163, 345
 - parmlib options information 333
 - pool details 188
 - rack number details 163, 349
 - rack number list 172, 372
 - scratch volume 28, 147, 326
 - shelf location details 163, 331
 - shelf location list 172, 358
 - software product details 163, 347
 - software product list 172, 369
 - vital record specification details 163, 355
 - vital record specification list 172, 392
 - volume details 163, 351
- requesting (*continued*)
 - volume list 163, 376
 - volume movements and actions list 190
 - volume release 37
- reserved words
 - ABEND 102
 - OPEN 101
- resetting in transit flag 306
- resetting the ABEND flag 299
- resource
 - defining 53
 - information 163
 - lists 172
 - system options 171
- restarting DFSMSrmm subsystem 193, 194
- restricted operands or fields 270, 282
- retaining
 - based on catalog status 100
 - by job name 99, 102
 - by last referenced date 100
 - data set retention types 6
 - data sets 5, 99
 - types of retention 5
 - until expired 100, 101
 - volumes 99
 - by generic volume serial number 90
 - by specific volume serial number 90
 - using GDG name 88
 - with special dates 83
 - volumes with special dates 104
 - when closed during ABEND processing 101
 - while cataloged 100
 - while closed during ABEND processing 101
 - while open 101
- retaining a data set using a vital record specification management value 139
- retaining data sets as long as they are cataloged 140
- retaining data sets based on expiration date 140
- retaining data sets based on the last referenced date 138
- retaining generation data group data sets 139
- retention date
 - calculating 106
 - listing 351
 - obtaining information 386
- retention period 103, 154
- retention policies 99
- retention types 5
- return codes
 - ADDBIN subcommand 226
 - ADDDATASET subcommand 232
 - ADDOWNER subcommand 235
 - ADDPRODUCT subcommand 237
 - ADDRACK subcommand 240
 - ADDVOLUME subcommand 254
 - ADDVRS subcommand 269
 - CHANGEDATASET subcommand 276
 - CHANGEOWNER subcommand 279
 - CHANGEPRODUCT subcommand 281
 - CHANGEVOLUME subcommand 307
 - DELETEDBIN subcommand 310

- return codes (*continued*)
 - DELETEDASET subcommand 226
 - DELETEOWNER subcommand 313
 - DELETEPRODUCT subcommand 315
 - DELETERACK subcommand 319
 - DELETEVOLUME subcommand 322
 - DELETEVRS subcommand 325
- description 401
- GETVOLUME subcommand 330
- LISTCONTROL subcommand 341
- LISTDATASET subcommand 344
- LISTOWNER subcommand 345
- LISTPRODUCT subcommand 348
- LISTRACK subcommand 332, 350
- LISTVOLUME subcommand 354
- LISTVRS subcommand 357
- SEARCHBIN subcommand 361
- SEARCHDATASET subcommand 368
- SEARCHPRODUCT subcommand 371
- SEARCHRACK subcommand 375
- SEARCHVOLUME subcommand 391
- SEARCHVRS subcommand 400
- RETURN command 51
- return to owner release action 34, 151
- return to scratch release action 34, 150
- reversing movement and action confirmation 284
- REXX exec
 - creating 411
 - EDGXMP1 VOLCHAIN EXEC 426
 - EDGXMP2 DSNLIST EXEC 429
 - testing return codes 401
 - using TRACE in 40
 - variables used 411
- rules for tape mount validation 17
- running mode
 - description 17

S

- SAF (System Authorization Facility)
 - description 16
- SAVE command 43
- scratch pool
 - converting from closed-cycle gdgs 112
 - definition 9
 - volume
 - adding 59
 - assigning 28, 147
 - defining 57
 - list 185
 - requesting 28, 147
- scratch volume
 - assigning 326
 - requesting 326
 - searching for information about 366
- search limit 172
- SEARCHBIN Subcommand
 - overview 358
 - return codes 361
 - syntax 359

- SEARCHDATASET Subcommand
 - overview 362
 - return codes 368
 - syntax 363
- searching
 - for data sets in the same multi-volume data set 363, 381
 - for data sets marked open 363, 380, 385
- SEARCHPRODUCT Subcommand
 - overview 369
 - return codes 371
 - syntax 370
- SEARCHRACK Subcommand
 - overview 372
 - return codes 375
 - syntax 373
- SEARCHVOLUME Subcommand
 - overview 376
 - return codes 391
 - syntax 378
- SEARCHVRS Subcommand
 - overview 392
 - return codes 400
 - syntax 394
- security class masks 339
- selecting command processing 43
- sequence number 74, 164
- session, DFSMSrmm
 - starting 23, 39
 - stopping 24, 50
- setting
 - date format 42
 - expiration date 103
 - home location 113
 - maximum retention period 11
 - owner notification 64
 - policy requirements 38
 - setting a default vital record specification 82
 - volume expiration date 11
- shelf location
 - adding information 53, 224, 238
 - changing 122
 - defining 53, 224, 238
 - deleting information 55, 308, 317
 - details 163, 331, 349
 - line operators 182, 184
 - list 172, 358, 372
 - management 9
- shelf-management
 - built-in storage location 3
 - description 9
 - installation defined storage location 3
- shutting down the DFSMSrmm subsystem
 - operator system error message issued 206
 - process for 195
- software product
 - adding information 66, 236
 - changing information 69, 280
 - defining 66, 236
 - deleting information 71, 315
 - details 163, 347

- software product (*continued*)
 - line operators 66, 184
 - list 172, 369
 - updating information 69, 280
 - volume 68, 71
- SORT command
 - description 46
 - using 179
- sorting
 - changing sort order 26
 - default sort order 25
 - directions 46
 - lists 25, 44, 176
 - options 44, 176
 - order 176
 - priorities 46
- special data set name mask 83
- special dates, support for 104
- stacked physical volume, changing 305
- START command 193
- starting
 - DFSMSrmm dialog 23, 39
 - DFSMSrmm subsystem 193
- starting a search for volume information 383
- status for a volume 57
- STGADMIN.EDG.IGNORE.TAPE 60
- STGADMIN.EDG.LISTCONTROL 171, 333
- STGADMIN.EDG.MASTER 60
- STGADMIN.EDG.RELEASE 320
- sticky label
 - default label description 218
 - processing 217
- stopping
 - DFSMSrmm dialog 24, 50
 - DFSMSrmm subsystem 195
- storage administrator
 - tasks 20
- storage class recorded by DFSMSrmm 72
- storage group recorded by DFSMSrmm 72
- storage location
 - adding bin number 54
 - assigning bin numbers 9
 - built-in 3
 - definition 3
 - deleting bin number 55
 - DISTANT 3
 - installation defined 3
 - LOCAL 3
 - moving from 123
 - moving volumes 99
 - REMOTE 3
 - retaining volumes 99
 - shelf locations 53
 - shelf-management 3
- submitting
 - batch job 51, 222
- SUPPORT fast path command 433
- SUPPORT operand 40
- system error messages 206
- system-managed tape library
 - adding volumes example 254
- system-managed tape library (*continued*)
 - description 254
 - ejecting volumes 120, 388
 - freeing up space 114
 - moving volumes 99, 114
 - moving volumes to 120
 - obtaining information about resident volumes 57
- system options 171
- system programmer
 - tasks 20

T

- table field name 25
- tape
 - processing tape labels 17
 - validating mounts 17
- tape configuration data base
 - definition 2
 - obtaining information from 57
 - updating volume information 120
 - validating library name 120
- tape label
 - expiration date 154
 - processing 19
 - specifying 247, 383
 - type 247, 383
 - types supported 12
 - validation 19, 120
- tape librarian
 - deleting an owner ID 65
 - DFSMSrmm ISPF Librarian Menu 22
 - tasks 20
- tape mount messages 200
- tape mount validation rules 17
- tape processing messages 205
- tape validation messages 202
- tasks
 - general user 20
 - operator 21
 - storage administrator 20
 - system programmer 20
 - tape librarian 20
- testing
 - return codes in a CLIST 401
 - return codes in a REXX procedure 401
- TRACE operand 40
- tracking I/O errors 150
- transferring volume ownership 66, 313
- trial run processing 8
- TSO subcommand
 - abbreviations 221
 - ADDBIN subcommand 224
 - ADDDATASET subcommand 227
 - ADDOWNER subcommand 233
 - ADDPRODUCT subcommand 236
 - ADDRACK subcommand 238
 - ADDVOLUME subcommand 241
 - ADDVRS subcommand 255
 - batch job 222
 - CHANGEDATASET subcommand 270

TSO subcommand *(continued)*

- CHANGEOWNER subcommand 221
- CHANGEPRODUCT subcommand 280
- CHANGEVOLUME subcommand 282
- DELETEBIN subcommand 308
- DELETEDATASET subcommand 311
- DELETEOWNER subcommand 313
- DELETEPRODUCT subcommand 315
- DELETERACK subcommand 317
- DELETEVOLUME subcommand 320
- DELETEVRS subcommand 324
- GETVOLUME subcommand 326
- LISTBIN subcommand 331
- LISTCONTROL subcommand 333
- LISTDATASET subcommand 342
- LISTOWNER subcommand 345
- LISTPRODUCT subcommand 347
- LISTRACK subcommand 349
- LISTVOLUME subcommand 351
- LISTVRS subcommand 355
- online help 222
- return codes 401
- SEARCHBIN subcommand 358
- SEARCHDATASET subcommand 362
- SEARCHPRODUCT subcommand 369
- SEARCHRACK subcommand 372
- SEARCHVOLUME subcommand 376
- SEARCHVRS subcommand 392
- specifying 221
- using 221
- variables 415
- types of retention 99

U

- uncataloging a data set 77
- undoing movement and action confirmation 284
- until expired
 - closed data sets retained 101
 - data sets retained until expired 100, 101
 - data sets retained while open 101
 - merging UNTILEXPIRED and WHILECATALOG 105
 - relationship with COUNT and WHILECATALOG operands 101
 - retention by open status 101
 - retention by volume expiration date 101
- updating
 - data set information automatically 35
 - data set information with RMM CHANGEDATASET subcommand 73, 270
 - owner information 31, 64, 277
 - software product information 69, 280
 - vital record specification 94
 - volume information 31, 60, 282
- user, general 23
- USER fast path command 433
- user menus 47
- USER operand 40
- user status 9
- user volume 57
 - definition 57

user volume 57 *(continued)*

- overwriting 57
- searching for information about 366
- using the action bar 46
- using the EDGUX100 installation exit 60
- utility
 - EDGAUD, security and audit 8
 - EDGBKUP, backing up control data set 8
 - EDGHSKP, inventory management 8
 - EDGINERS, initializing and erasing volumes 8
 - EDGRPTD, movement and inventory 15
 - EDGUTIL, verifying control data set contents 8

V

- validating
 - library name 120
 - magnetic tape mounts 17
- variables, TSO subcommand 415
- vaulting support for logical volumes 143
- virtual tape server
 - adding an exported logical volume 246
 - building an export/import list 389
 - changing logical volume location 295
 - changing volume type 305
 - completing the import process 145
 - confirming volume moves for exported volumes 144
 - creating a volume export list 143
 - creating a volume import list 145
 - description 2
 - ejecting a logical volume 295
 - export processing 143
 - identifying logical volumes to DFSMSrmm 389
 - import processing 144
 - volume not in library support 144
- vital record group 81
- vital record specification
 - ABEND 102
 - chaining 7, 79
 - changing 94
 - creating list 172, 392
 - data set
 - adding 92, 255
 - changing 95
 - deleting 324
 - details 163, 355
 - list 172, 392
 - retention 103
 - defining 90, 255
 - deleting 96, 324
 - examples
 - chaining vital record specifications 80
 - line operators 182, 184
 - linking 255
 - managing vital records 99
 - matching data set names 84, 85
 - matching job names 84, 85
 - matching order 84, 85
 - matching types 342
 - merging UNTILEXPIRED and WHILECATALOG retention policies 105

vital record specification (*continued*)

- name
 - adding 94, 255
 - deleting 97, 324
 - details 163, 355
 - list 392
- OPEN 101
- policy requirements 38
- requesting information 163, 355
- retaining DFSMSHsm ABARS backups 268
- retaining DFSMSHsm backup and migration tapes 256
- setting an installation default 82
- updating 94
- using 99
- volume
 - adding 93, 255
 - deleting 324
 - details 163, 355
 - list 172, 392
 - retention 103
- vital record specification management value
 - assigned by DFSMSrmm EDGUX100 sample 6
 - defining management and retention policy 255
 - description 6
 - displaying 342
 - matching with data set names 84, 85
 - specifying 104, 262
- VOL1, clearing the label information 304
- VOL1, searching for information 386
- VOL1 label, specifying ISO/ANSI label version 251
- VOLCHAIN EXEC 426
- volume
 - actions list 190
 - adding information 57, 59, 241
 - automatic recording of information 56
 - canceling outstanding moves 285
 - changing 285, 299
 - changing information 282
 - changing volume type 305
 - clearing location information 295
 - confirming moves 124, 186
 - creating list 376
 - date assigned to a user 245
 - defining 56, 241
 - deleting information 62, 320
 - details 163, 351
 - ejecting 120
 - eligible for release 148
 - erasing 152, 208
 - grouping 38
 - high security 152
 - in transit 124
 - initial volume serial number 59
 - initializing 152, 208
 - limiting the search for volume information 383
 - line operators 182, 184, 191
 - list 172
 - loan location 122
 - manually releasing 29, 158
 - master status 9, 57

volume (*continued*)

- movements list 190
- moving 99, 113
- non-restricted operands or fields 282
- non-scratch
 - adding 57
 - changing information 31, 60
- obtaining information from the TCDB 57
- overwriting
 - master volumes 57
 - user volumes 57
- private 9, 57
- reclaiming from pending status or scratch status 283
- redefining duplicate volume serial number 60
- release actions 34, 148
- release date 34, 154
- releasing 148, 320
- replacing 150
- requesting movement 118
- requirements for pooling 38
- resetting the ABEND status flag 299
- restricted operands or fields 282
- retained by job name 102
- retained until expired 100, 101
- retained while cataloged 100
- retained while open 101
- retaining
 - by generic volume serial number 90
 - by specific volume serial number 90
- retention date 109
- retention types 5
- returning from a loan location 123
- returning to home location 163
- returning to owner 151
- returning to scratch status 150
- scratch volume 9, 57
- software product 68, 71
- status 57
- tracking I/O errors 150
- transferring ownership 66
- types of movement 99
- updating information 282
- user 57
- user status 9
- using CHANGEVOLUME subcommand to confirm volume movement 284
- validating mounts 17
 - with permanent I/O errors 150
 - with the same volume serial number 60
- volume catalog
 - obtaining information about volumes 254
- VOLUME fast path command 433
- volume not in library support 144
- VOLUME operand 40
- volume serial number
 - definition 90
 - external 2
 - generic volume serial number 90
- volume vital record specification
 - adding 93

- volume vital record specification *(continued)*
 - definition 93
- VRS fast path command 433
- VRS operand 40
- VRSBYJOBNAME
 - defining a vital record specification 102
 - defining job names 85
 - description 5
 - filtering with job name masks 87

W

- while cataloged
 - catalog date format 109
 - data sets retained while cataloged 100
 - merging WHILECATALOG and UNTILEXPIRED 105
 - minimum retention with WHILECATALOG 101, 267
 - vital record processing for 105
 - with special dates 268
- wrong data set name, changing 274
- wrong label processing 211
- wrong volume serial number, changing 299

Readers' Comments — We'd Like to Hear from You

DFSMS/MVS Version 1 Release 5
DFSMSrmm Guide and Reference

Publication No. SC26-4931-06

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



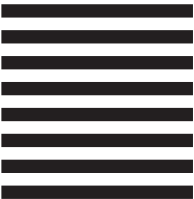
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
RCF Processing Department
G26/050
5600 Cottle Road
SAN JOSE, CA 95193-0001



Fold and Tape

Please do not staple

Fold and Tape



File Number: S370/S390-40
Program Number: 5695-DF1
5647-A01



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-4931-06



Spine information:



DFSMS/MVS Version 1 Release 5 DFSMSrmm Guide and Reference