z/OS

DFSMSrmm Guide and Reference

SC26-7404-04

z/OS

DFSMSrmm Guide and Reference

SC26-7404-04

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 423.

Fifth Edition, September 2004

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

This edition replaces SC26-7404-03.

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

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

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

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

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

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

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

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

© Copyright International Business Machines Corporation 1992, 2004. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures xi

Tables xv

About This Document xvii
Required product knowledge xvii
Referenced documents xvii
Accessing z/OS DFSMS documents on the Internet xvii
Using LookAt to look up message explanations xviii
Accessing z/OS licensed documents on the Internet. xviii
Notational conventions xix
How to read syntax diagrams xix
How to abbreviate commands and operands xxi
How to use continuation characters xxi
Delimiters xxii
Character sets xxii

Summary of Changes xxv
Summary of Changes for SC26-7404-04 z/OS Version 1 Release 6 xxv
New Information xxv
Changed Information xxv
Moved Information xxvi
Summary of Changes for SC26-7404-03 z/OS Version 1 Release 5 xxvi
New Information xxvi
Changed Information xxvi
Summary of Changes for SC26-7404-02 z/OS Version 1 Release 3 xxvi
New Information xxvi
Changed Information xxvii
Summary of Changes for SC26-7404-01 z/OS Version 1 Release 3. xxvii
New Information. xxvii
Deleted Information xxvii
Changed Information xxvii
Moved Information xxvii

Chapter 1. Getting Started with DFSMSrmm 1
Using the Dialog 1
Starting the Dialog 1
User Menus 4
Command Menu 4
Function Menus 5
Getting Information about Your Resources 5
Changing Options 6
Displaying Options 9
Navigating Through DFSMSrmm 9
Issuing Fast Path Commands 10
Requesting Help 11
Exiting the Dialog 11
Using Lists 12
Issuing the SORT Command 12
Using SORT Options Panels 13
Changing the Default Sort Order 14
Printing Display Lists and Panels 15
Requesting Batch Details 15

© Copyright IBM Corp. 1992, 2004

Using Volumes 16
Requesting a Scratch Volume 16
Changing Volume Information 18
Changing the Release Date for a Volume 18
Setting Release Actions for a Volume 19
Changing Owner Information 19
Releasing Volumes Manually 19
Requesting Notification of Release 20
Requesting Return after Release 20
Retaining Data Sets and Volumes 20
Grouping Volumes 21
Changing Data Set Information 21

Chapter 2. Defining Your Resources Using the DFSMSrmm ISPF Dialog 23
Defining Shelf Locations 23
Adding New Rack and Bin Numbers 24
Deleting Rack and Bin Numbers 25
Defining Volumes 25
Specifying Media Type and Recording Format for Volumes 26
Adding a New, Non-Scratch Volume 27
Adding Scratch Volumes 28
Adding Volumes with Duplicate Volume Serial Numbers 28
Redefining a Volume Already Defined to DFSMSrmm 29
Changing Volume Information 29
Changing Information for Volumes in Multivolume Sets 30
Deleting Volume Information 31
Deleting Information for Volumes in Multivolume Sets 31
Defining Owners 31
Adding New Owners 31
Changing Owner Information 32
Deleting Owner Information 32
Defining Software Products and Product Versions 33
Adding a New Software Product 33
Adding Software Product Volumes 33
Changing Software Product Information 34
Deleting Software Product Information 34
Defining Data Sets 35
Adding New Data Set Information 35
Changing Data Set Information 36
Deleting Data Set Information 36

Chapter 3. Defining Vital Record Specifications 39
Types of Vital Record Specifications 39
Data Set Vital Record Specifications 39
Name Vital Record Specifications 40
Volume Vital Record Specifications 40
Chaining Vital Record Specifications 40
Identifying Data Sets for Retention 41
Fully Qualified Data Set Names 41
Data Set Name Masks 42
Management Class and Management Value Names 43
Special Data Set Name Masks 43
Name Filtering with Data Set Name Masks 43
How DFSMSrmm Applies Retention and Movement Policies 43
Generation Data Group (GDG) Base Names 48
Pseudo-GDG Data Set Names 48

Specifying GDG and Non-GDG Data Set Names 49
Identifying Volumes for Retention 50
Retaining Volumes by Specific Volume Serial Number 50
Retaining Volumes by Generic Volume Serial Number 50
Adding Vital Record Specifications Using the DFSMSrmm ISPF Dialog 51
Adding Data Set Vital Record Specifications 51
Adding Volume Vital Record Specifications 51
Adding Name Vital Record Specifications 51
Changing Vital Record Specifications. 52
Deleting a Vital Record Specification 52

Chapter 4. Retaining and Moving Your Volumes 55
Types of Retention 55
CYCLES 55
BYDAYSCYCLE 56
DAYS 56
EXTRADAYS 56
LASTREFERENCEDAYS 56
WHILECATALOG 56
UNTILEXPIRED 57
By Data Set Open Status 57
By Job Name 58
By Specific Volume Serial Number. 58
By Generic Volume Serial Number. 58
By Specific Date 58
Defining Retention Policies for Data Sets and Volumes 58
Defining Policies for Management Class and Management Value 59
Combining Retention Policies in a Vital Record Specification 60
Merging Retention Policies 61
Retaining and Moving Volumes as Sets or Individually 61
Defining Release Policies 62
Defining System-Wide Vital Record Specifications 62
DFSMSrmm Retention Dates. 63
Retention Date Format 63
Special Cycles Date Format 64
Special Catalog Date Format. 64
Special CATRETPD Date Format 64
How DFSMSrmm Calculates the Retention Date 65
Data Set Retention Date 65
Volume Retention Date 65
Retaining Data Sets in Closed-Cycle GDGs 69
Managing Closed-Cycle GDGs 69
Converting from Closed-Cycle GDGs to Using Scratch Pools 70
Defining Movement Policies for Volumes 70
Moving Physical Volumes between System-Managed Libraries 71
Managing the Contents of System-Managed Libraries 71
Moving to Storage Locations 73
Using Manual Move Control 73
Manually Requesting Volume Moves 74
Requesting a Single Volume Move 75
Moving Volumes to a System-Managed Tape Library 75
Ejecting Volumes from a System-Managed Tape Library. 76
Moving Volumes Between Pools 77
Moving Volumes to Loan Locations 78
Returning Volumes from Loan Locations 79
Moving Volumes from Storage Locations 79

Confirming Volume Movements to DFSMSrmm 79
Confirming Volume Moves into an Automated Tape Library 80
Confirming Volume Moves for Other Locations 80

Chapter 5. Defining Retention and Movement Policies 85
Before You Start Defining Retention and Movement Policies 85
A Summary of the Tasks To Perform 85
Where to Find Information 85
Which Type of Vital Record Processing Should You Use? 86
What is Your Level of Experience with DFSMSrmm? 86
Selecting VRSEL(OLD) Processing 87
Selecting VRSEL(NEW) Processing 87
Migrating to Using VRSEL(NEW) 88
EDGRVCLN REXX Procedure to Clean Up Name Vital Record Specifications 89
Chaining Retention and Movement Policies 91
What You Can Chain When You Specify VRSEL(OLD) 92
What You Can Chain When You Specify VRSEL(NEW) 92
Specifying Policies 92
Defining the Minimum Required Policy 94
Writing Your Own Vital Record Specifications 95
Example 1: Retaining Catalogued Data Sets 95
Example 2: Retaining Uncatalogued Data Sets 95
Example 3: Retaining Cycles of Data Sets 95
Example 4: Moving Data Sets to Different Locations 96
Example 5: Retaining Data Sets Created within a Time Period 96
Example 6: Holding Data Sets for Extra Days 96
Example 7: Retaining Generation Data Group Data Sets 97
Example 8: Retaining Recently Used Data Sets 97
Example 9: Using Management Value to Retain Data Sets 98
Example 10: Returning Volumes to Scratch Status 98
Example 11: Retaining Data Sets Using Expiration Date 98
Example 12: Combining Vital Record Specifications 99

Chapter 6. Requesting and Releasing Volumes 101
Requesting Scratch Volumes Manually 101
Releasing Volumes 102
Setting Release Actions for a Volume 102
Changing a Volume's Release Date 106
Confirming Manual Release Actions to DFSMSrmm 107
Releasing Volumes Manually 109

Chapter 7. Requesting Information about Your Resources 111
Using DFSMSrmm Reporting Utilities and Sample Reports 111
Displaying Details about a Single Resource 111
Displaying DFSMSrmm System Options 113
Using Resource Lists 114
How to Request a List 115
How to Sort a List 117
How to Use Line Operators 123
Requesting Specific Types of Lists 124
Scratch Volume Pull Lists 124
Dropship Lists 126
Lists of Pools Defined by Your Installation 127
Lists of Volume Movements and Actions 127
Creating CLISTs of Executable Subcommands 128
Printing Resource Information 129

Chapter 8. DFSMSrmm Operator Procedures 131
Restarting the DFSMSrmm Subsystem 131
Using the START Command 131
Using the Modify Command. 132
Stopping the DFSMSrmm Subsystem 134
Controlling the Problem Determination Aid (PDA) Facility 136
Responding to DFSMSrmm Operator Messages 136
Initialization Messages 137
Tape Mount Messages 140
Tape Validation Messages 142
Tape Processing Messages. 145
System Error Messages 146
DFSMSrmm Installation Exit Messages 148
Initializing and Erasing Tape Volumes Manually 149
Using the LABEL Procedure to Request EDGINERS Processing 149
Replying to LABEL Procedure Messages 155
INIT and ERASE Request Syntax 157
Processing Sticky Labels. 160
Labels 160

Chapter 9. Using RMM TSO Subcommands 163
Issuing the RMM TSO Command and Subcommands 163
Requesting Help for RMM TSO Subcommands 164
Submitting a Batch Job 165
ADDBIN: Adding a Bin Number in a Storage Location 166
ADDDATASET: Adding Data Set Information 169
ADDOWNER: Adding Owner Information 176
ADDPRODUCT: Adding Software Product Information 179
ADDRACK: Adding a Shelf Location 182
ADDVOLUME: Adding Volume Information 186
ADDVRS: Adding a Vital Record Specification 205
CHANGEDATASET: Changing Data Set Information. 219
CHANGEOWNER: Changing Owner Information 229
CHANGEPRODUCT: Changing Software Product Information 232
CHANGEVOLUME: Changing Volume Information 234
DELETEBIN: Deleting Bin Number Information. 263
DELETEDATASET: Deleting Data Set Information 266
DELETEOWNER: Deleting Owner Information 268
DELETEPRODUCT: Deleting Software Product Information 270
DELETERACK: Deleting Shelf Location Information 272
DELETEVOLUME: Deleting Volume Information 275
DELETEVRS: Deleting Vital Record Specifications 279
GETVOLUME: Requesting and Assigning Scratch Volumes 281
LISTBIN: Displaying Information about a Shelf Location 285
LISTCONTROL: Displaying Parmlib Options and Control Information 287
LISTDATASET: Displaying Information about a Data Set 295
LISTOWNER: Displaying Information about an Owner 298
LISTPRODUCT: Displaying Information about a Software Product. 300
LISTRACK: Displaying Information about a Shelf Location 302
LISTVOLUME: Displaying Information about a Volume 304
LISTVRS: Displaying Information about a Vital Record Specification 307
SEARCHBIN: Creating a List of Bin Numbers 310
SEARCHDATASET: Creating a List of Data Sets 315
SEARCHPRODUCT: Creating a List of Software Products 322
SEARCHRACK: Creating a List of Shelf Locations 325
SEARCHVOLUME: Creating a List of Volumes. 329

SEARCHVRS: Creating a List of Vital Record Specifications. 347

Chapter 10. DFSMSrmm Return Codes and Reason Codes 355
RMM TSO Subcommand Return Codes 355
RMM TSO Subcommand Reason Codes 356

Chapter 11. DFSMSrmm REXX Variables 363
TSO Subcommand Variables by Subcommand. 363
TSO Subcommand Variables by Name 367

Appendix A. DFSMSrmm TSO Subcommands 381
ADDBIN: Adding a Bin Number in a Storage Location 381
ADDDATASET: Adding Data Set Information 382
ADDOWNER: Adding Owner Information 384
ADDPRODUCT: Adding Software Product Information 384
ADDRACK: Adding a Shelf Location 385
ADDVOLUME: Adding Volume Information 386
ADDVOLUME: Volume Operands 386
ADDVOLUME: Non-SCRATCH Volume Optional Operands 387
ADDVRS: Adding a Vital Record Specification 390
ADDVRS: Adding a Data Set Vital Record Specification 390
ADDVRS: Adding a Location Name Vital Record Specification 391
ADDVRS: Adding a Retention Name Vital Record Specification 392
ADDVRS: Adding a Volume Vital Record Specification 394
CHANGEDATASET: Changing Data Set Information. 395
CHANGEOWNER: Changing Owner Information 397
CHANGEPRODUCT: Changing Software Product Information 397
CHANGEVOLUME: Changing Volume Information 398
CHANGEVOLUME: Confirmrelease and Global Movement Operands . . . 398
CHANGEVOLUME: Specific Volume Optional Operands 399
DELETEBIN: Deleting Bin Number Information. 404
DELETEDATASET: Deleting Data Set Information 404
DELETEOWNER: Deleting Owner Information 404
DELETEPRODUCT: Deleting Software Product Information 405
DELETERACK: Deleting Shelf Location Information 405
DELETEVOLUME: Deleting Volume Information 406
DELETEVRS: Deleting Vital Record Specification Information 406
GETVOLUME: Requesting and Assigning Scratch Volumes 407
LISTBIN: Displaying Information about a Shelf Location 408
LISTCONTROL: Displaying Parmlib Options and Control Information 408
LISTDATASET: Displaying Information about a Data Set 408
LISTOWNER: Displaying Information about an Owner 408
LISTPRODUCT: Displaying Information about a Software Product. 409
LISTRACK: Displaying Information about a Shelf Location 409
LISTVOLUME: Displaying Information about a Volume 409
LISTVRS: Displaying Information about a Vital Record Specification 409
SEARCHBIN: Creating a List of Bin Numbers 410
SEARCHDATASET: Creating a List of Data Sets 411
SEARCHPRODUCT: Creating a List of Software Products 412
SEARCHRACK: Creating a List of Shelf Locations 412
SEARCHVOLUME: Creating a List of Volumes. 413
SEARCHVRS: Creating a List of Vital Record Specifications. 416

Appendix B. DFSMSrmm ISPF Dialog Fast Path Commands 417

Appendix C. Accessibility 421

Using assistive technologies 421
Keyboard navigation of the user interface. 421
z/OS information 421

Notices 423
Programming interface information 424
Trademarks. 424

Glossary 425

Index 437

Figures

1. DFSMSrmm Primary Option Menu . . . 1
2. DFSMSrmm User Menu. . . . 2
3. RMMISPF Exec Syntax Diagram . . . 3
4. DFSMSrmm Command Menu . . . 4
5. DFSMSrmm Volume Menu. . . . 5
6. DFSMSrmm Dialog User Options Panel . . . 7
7. DFSMSrmm Dialog Options Menu . . . 9
8. DFSMSrmm Volume Display Panel . . . 10
9. DFSMSrmm Exit Menu . . . 12
10. DFSMSrmm Dialog Sort Options Menu. . . . 13
11. DFSMSrmm Data Set List Sort Options Panel . . . 14
12. DFSMSrmm Volume List Sort Options Panel . . . 15
13. DFSMSrmm Processing Multiple Volumes . . . 30
14. Chaining Vital Record Specifications . . . 40
15. Specifying a Name Vital Record Specification for a Chain . . . 41
16. Specifying a Vital Record Specification Chain . . . 41
17. Valid Job Name Masks . . . 46
18. Defining a Retention Policy for a GDG Base Name . . . 48
19. Specifying Data Set Name Masks for Pseudo-GDGs. . . . 49
20. Specifying Data Set Name Masks to Manage Data Sets Like a GDG . . . 49
21. Specifying A Global Data Set Name Mask to Manage GDGs. . . . 49
22. Specifying NOGDG and CYCLES in a Vital Record Specification . . . 50
23. Specifying NOGDG and DAYS in a Vital Record Specification . . . 50
24. Retaining a Data Set Using Management Class . . . 60
25. Retaining a Data Set Using Management Value . . . 60
26. Specifying Combinations of Retention Policies on a Single Vital Record Specification . . . 61
27. Using UNTILEXPIRED and WHILECATALOG on a Single Vital Record Specification . . . 61
28. Merging UNTILEXPIRED with the WHILECATALOG Retention Policy . . . 61
29. Managing Closed-Cycle GDGs. . . . 70
30. Managing Volumes Using Scratch Pools . . . 70
31. Moving Volumes out of a System-Managed Library . . . 72
32. Moving a Volume in and out of the System-Managed Library. . . . 72
33. Moving Data out of the System-managed Library at Fixed Intervals . . . 73
34. Keeping Volumes On-site. . . . 74
35. Sending a Volume to Another System . . . 74
36. Returning a Volume from Another System . . . 74
37. Ejecting a Volume from a System-managed Library . . . 77
38. Moving a Volume to a Loan Location . . . 78
39. Supplying Loan Location Information for Volumes . . . 79
40. Returning a Volume from a Loan Location . . . 79
41. DFSMSrmm Volume Action Status Panel . . . 81
42. DFSMSrmm Volume Action Summary List . . . 82
43. Confirming Volume Moves for Volumes Ready to Scratch . . . 83
44. Identifying Locations and Destinations for Volumes . . . 83
45. Sample JCL to Run the EDGRVCLN Procedure . . . 91
46. Specifying a Vital Record Specification With Default Operands . . . 93
47. Specifying a Vital Record Specification Using the Dialog Part 1. . . . 93
48. Specifying a Vital Record Specification Using the Dialog Part 2. . . . 94
49. Confirming Replace Actions for Volumes. . . . 109
50. DFSMSrmm Volume Display Panel. . . . 113
51. DFSMSrmm System Options Display . . . 114
52. DFSMSrmm Dialog Sort Options Menu . . . 117
53. DFSMSrmm Data Set List Sort Options Panel. . . . 118

© Copyright IBM Corp. 1992, 2004

54. SORT Command Syntax Diagram for Rack or Bin Number Lists . . . 121
55. SORT Command Syntax Diagram for Volume Lists . . . 122
56. SORT Command Syntax Diagram for Lists of Volume Movements and Actions . . . 122
57. SORT Command Syntax Diagram for Software Product . . . 122
58. SORT Command Syntax Diagram for Data Set Lists . . . 123
59. SORT Command Syntax Diagram for Rack or Bin Number Lists . . . 123
60. SORT Command Syntax Diagram for Report Generator Lists . . . 123
61. Producing a List of Scratch Volumes . . . 125
62. DFSMSrmm Volume Search Panel . . . 125
63. DFSMSrmm Volume Action Status Panel . . . 126
64. START Command Syntax Diagram . . . 131
65. MODIFY Command Syntax Diagram . . . 132
66. LABEL Command Syntax Diagram . . . 150
67. INIT and ERASE Request Syntax Diagram . . . 158
68. Default Label Format for a Tape Cartridge . . . 161
69. Default Label Format for a Round Tape . . . 161
70. Sample LISTBIN Output . . . 286
71. Sample LISTCONTROL Output . . . 290
72. Sample LISTCONTROL Output . . . 291
73. Sample LISTCONTROL Output . . . 292
74. LISTCONTROL LOCDEF Output - No LOCDEFs Defined . . . 293
75. LISTCONTROL LOCDEF Output - LOCDEFs Defined . . . 294
76. Sample LISTDATASET Output . . . 296
77. Sample LISTOWNER Output . . . 298
78. Sample LISTPRODUCT Output . . . 301
79. Sample LISTRACK Output . . . 303
80. Sample LISTVOLUME Output . . . 305
81. Sample LISTVOLUME Output . . . 306
82. Listing Vital Record Specification Information Examples . . . 308
83. Sample LISTVRS Output . . . 309
84. SEARCHBIN Syntax Diagram . . . 311
85. Sample SEARCHBIN Output for Empty Bins . . . 313
86. Sample SEARCHBIN Output . . . 313
87. Sample SEARCHDATASET Output . . . 320
88. Sample SEARCHDATASET Output . . . 320
89. Sample SEARCHDATASET Output . . . 320
90. Sample SEARCHDATASET Output . . . 321
91. Sample CLIST Data Set . . . 321
92. Sample SEARCHPRODUCT Output . . . 324
93. Sample SEARCHRACK Output . . . 328
94. Sample SEARCHVOLUME Output Listing Duplicate Volumes . . . 341
95. Sample SEARCHVOLUME Output Listing Owned Volumes . . . 342
96. Sample SEARCHVOLUME Output Listing Volumes to Eject . . . 342
97. SEARCHVOLUME Output Using CLIST Operand . . . 342
98. Sample SEARCHVOLUME Output Using RETDATE Operand . . . 343
99. Sample SEARCHVOLUME Output Building a List of Logical Volumes . . . 343
100. Sample SEARCHVOLUME CLIST Output for a List of Logical Volumes . . . 343
101. Sample SEARCHVOLUME CLIST Output Identifying Logical Volumes to DFSMSrmm . . . 343
102. Moving Volumes from One Library to Another Library . . . 344
103. Confirming Volume Moves from One Library to Another Library . . . 344
104. Listing Volumes within a Range of Volume Serial Numbers . . . 344
105. Using a Generic Volume Serial Number to Obtain a List of Volumes within a Range of Volume Serial Numbers . . . 345
106. Listing Volumes for an Owner within a Range of Volume Serial Numbers . . . 345
107. Listing Volumes for the Export List . . . 345
108. CLIST Information for the Export List . . . 346

109. Sample SEARCHVRS Output. 352
110. Sample SEARCHVRS Output. 352
111. Sample SEARCHVRS Output. 353
112. Testing RMM TSO LISTVOLUME Subcommand Example 1 355
113. Testing RMM TSO LISTVOLUME Subcommand Example 2 355

Tables

1. Character Sets xxii
2. Special Characters Used in Syntax xxii
3. Requesting Lists and Display Panels from the User Menu 5
4. Date Format Options 7
5. DFSMSrmm Media Types 26
6. How DFSMSrmm Assigns Media Type and Recording Format 27
7. How DFSMSrmm Matches Data Set Names to Data Set Masks 44
8. How DFSMSrmm Uses Jobname to Apply Policies 47
9. How DFSMSrmm Matches Job Names to Job Name Masks 47
10. Date Format Displayed 64
11. DFSMSrmm Retention Date Calculation by COUNT from 1 through 99998 66
12. DFSMSrmm Retention Date Calculation by COUNT(99999) 67
13. Release Actions for a Volume. 103
14. Confirm List Line Operators 108
15. DFSMSrmm Release Volumes Line Operators 110
16. Information to Specify on Display Panels 112
17. How to Display Search Panels 115
18. Table Field Names on DFSMSrmm Lists. 118
19. Operator Messages 136
20. Operator Responses to Message EDG1107D 140
21. Operator Responses to Message EDG6663D 141
22. Operator Responses to Message EDG6663D 157
23. RMM TSO Subcommands 163
24. New Owner Information 178
25. Assigning Rack Numbers to Volumes that DFSMSrmm Manages 182
26. How DFSMSrmm Assigns Rack Numbers to a Volume 186
27. Non-restricted Volume Information 235
28. Volume Information That Can Be Changed Under Special Conditions 235
29. Volume Information for Volumes in Master and User Status. 235
30. Volume Information for All Volumes 235
31. Volume Information That Can Be Changed with CONTROL Access 236
32. LISTCONTROL Subcommand Output. 293
33. Information Returned by SEARCHBIN 310
34. Information Returned by SEARCHDATASET 315
35. Information Returned By SEARCHPRODUCT. 322
36. Information Returned by SEARCHRACK 325
37. Information Returned by the RMM SEARCHVOLUME Subcommand 329
38. Creating Lists of Vital Record Specifications 347
39. Information Returned by SEARCHVRS 347
40. DFSMSrmm Return Codes. 355
41. DFSMSrmm Reason Codes 356
42. TSO Subcommand Variables by RMM Subcommand 363
43. TSO Subcommand Variables by Name 367
44. DFSMSrmm ISPF Dialog Fast Path Commands 417

© Copyright IBM Corp. 1992, 2004

About This Document

This document 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 document 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

For information about accessibility features of z/OS and z/OS.e, for users who have a physical disability, see Appendix C, “Accessibility,” on page 421.

Required product knowledge

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

- ISPF, for using the DFSMSrmm ISPF dialog
- TSO, for using the TSO command, RMM, and related subcommands

Referenced documents

The following documents have additional information about DFSMSrmm:

Document Title	Order Number
z/OS DFSMSrmm Diagnosis Guide	GY27-7619
z/OS DFSMSrmm Guide and Reference	SC26-7404
z/OS DFSMSrmm Implementation and Customization Guide	SC26-7405
z/OS DFSMSrmm Reporting	SC26-7406
z/OS DFSMSrmm Application Programming Interface	SC26-7403

This document also refers to the following documents:

Document Title	Order Number
z/OS MVS System Messages, Vol 3 (ASB-BPX)	SA22-7633
z/OS MVS System Messages, Vol 4 (CBD-DMO)	SA22-7634
z/OS MVS System Messages, Vol 5 (EDG-GFS)	SA22-7635

Accessing z/OS DFSMS documents on the Internet

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

© Copyright IBM Corp. 1992, 2004

Using LookAt to look up message explanations

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

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

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

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

Accessing z/OS licensed documents on the Internet

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

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

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

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

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

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

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

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

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

Printed licensed documents are not available from IBM.

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

Notational conventions

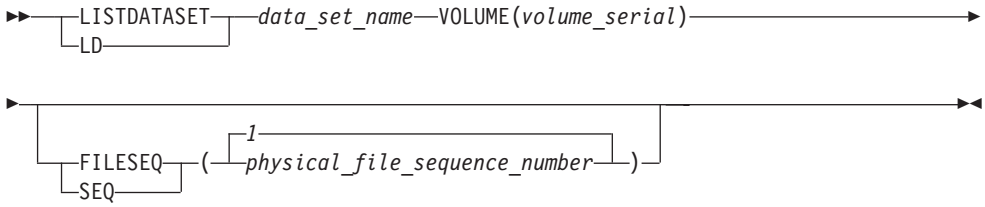
This section explains the notational conventions used in this document.

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.

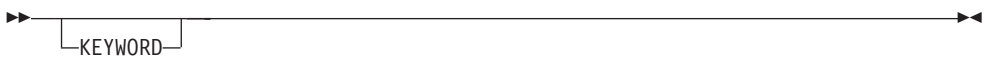
- If a diagram is longer than one line, each line to be continued ends with a single arrowhead and the next line begins with a single arrowhead.



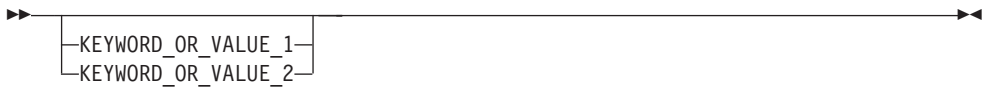
- Required keywords and values appear on the main path line. You must code required keywords and values.

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

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

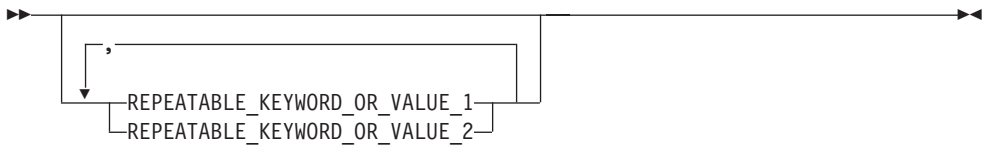


  If several mutually exclusive optional keywords or values exist, they are stacked vertically in alphanumeric order below the main path line.

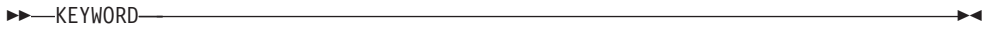


- An arrow returning to the left above a keyword or value on the main path line means that the keyword or value can be repeated. The comma means that each keyword or value must be separated from the next by a comma.

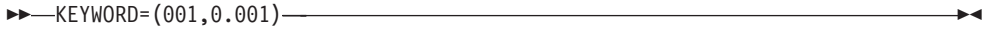
- An arrow returning to the left above a group of keywords or values means more than one can be selected, or a single one can be repeated.



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

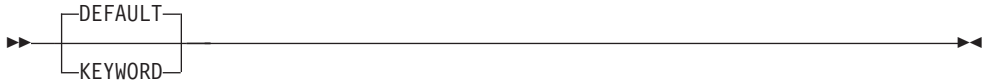


  If a keyword or value can be abbreviated, the abbreviation is discussed in the text associated with the syntax diagram.
- If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, 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.

Notes:

1 An example of a syntax note.

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

```
►►─KEYWORD──────────────────────────────────────────────────────────────►◄
```

- 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.

```
►►─┤ Reference to Syntax Fragment ├──────────────────────────────────────►◄
```

Syntax Fragment:

```
├─1ST_KEYWORD,2ND_KEYWORD,3RD_KEYWORD──────────────────────────────────┤
```

The following is an example of a syntax diagram.

```
►►─┬─DELETEOWNER─┬─owner_ID─┬──────────────┬─┬──────────────────────────►◄
   └─DO──────────┘          └─┤ newowner ├─┘
```

newowner

```
                        (1)
├─NEWOWNER(new_owner_ID)─┬──────────────────────────────────────────────┤
```

Notes:

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

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 document. 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.

Character sets

To code job control statements, use characters from the character sets in Table 1. Table 2 lists the special characters that have syntactical functions in job control statements.

Table 1. Character Sets

Character Set	Contents	
Alphanumeric	Alphabetic Numeric	Capital A through Z 0 through 9
National (See note)	“At” sign Dollar sign Pound sign	@ (Characters that can be $ represented by hexadecimal # values X'7C', X'5B', and X'7B')
Special	Comma Period Slash Apostrophe Left parenthesis Right parenthesis Asterisk Ampersand Plus sign Hyphen Equal sign Blank	, . / ' ( ) * & + - =
EBCDIC text	EBCDIC printable character set	Characters that can be represented by hexadecimal X'40' through X'FE'
Note: The system recognizes the following hexadecimal representations of the U.S. National characters; @ as X'7C'; $ as X'5B'; and # as X'7B'. In countries other than the U.S., the U.S. National characters represented on terminal keyboards might generate a different hexadecimal representation and cause an error. For example, in some countries the $ character may generate a X'4A'.		

Table 2. Special Characters Used in Syntax

Character	Syntactical Function
,	To separate parameters and subparameters
=	To separate a keyword from its value, for example, BURST=YES
(ḅ)	To enclose subparameter list or the member name of a PDS or PDSE
&	To identify a symbolic parameter, for example, &LIB
&&	To identify a temporary data set name, for example, &&TEMPDS, and, to identify an in-stream or sysout data set name, for example, &&PAYOUT
.	To separate parts of a qualified data set name, for example, A.B.C., or parts of certain parameters or subparameters, for example, nodename.userid

Table 2. Special Characters Used in Syntax (continued)

Character	Syntactical Function
*	To refer to an earlier statement, for example, OUTPUT=*.name, or, in certain statements, to indicate special functions: //label CNTL * //ddname DD * RESTART=* on the JOB statement
’	To enclose specified parameter values which contain special characters
(blank)	To delimit fields

Summary of Changes

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

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

Summary of Changes for SC26-7404-04 z/OS Version 1 Release 6

This document contains information previously presented in *z/OS Version 1 Release 5 DFSMSrmm Guide and Reference* (SC26-7404-03).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- The DFSMSrmm ISPF dialog has been enhanced for better usability. You can request that DFSMSrmm prime panels with saved variables. You can create a CLIST of DFSMSrmm TSO subcommands using the dialog. You can use new line operators in dialog lists to better manage DFSMSrmm resources. You can specify a new user option to define the type of output station where you want your system-managed cartridges ejected.
- DFSMSrmm can be used in a client/server environment. Part of this function includes new operator commands to stop and start the subsystem or IP tasks. This function also provides new REXX variables that you can use to write REXX execs to obtain information about your client/server environment.
- DFSMSrmm provides support for you to use the IBM TotalStorage Enterprise Tape System 3592 WORM tapes. Information has been added about the following topics.
  - Added new media types MEDIA6, MEDIA7, and MEDIA8.
  - Added operand values for ADDVOLUME, CHANGEVOLUME, and SEARCHVOLUME.
  - Added new return code and reason code
  - Added new REXX variables and updated several existing variables
  - Updated command syntax diagrams for ADDVOLUME, CHANGEVOLUME, and SEARCHVOLUME.
- The *z/OS DFSMSrmm Command Reference Summary* is no longer available as a separate document. The command summary information has been added to this book as an appendix.

Changed Information

The following information was moved in this edition:

- When you issue the RMM LISTCONTROL ACTIONS MOVES command, DFSMSrmm returns information from the control data set on the DFSMSrmm server system.

© Copyright IBM Corp. 1992, 2004

Moved Information

The following information was moved in this edition:

- The introductory information about DFSMSrmm has been moved to the *z/OS DFSMSrmm Implementation and Customization Guide*.

Summary of Changes for SC26-7404-03 z/OS Version 1 Release 5

This document contains information previously presented in *z/OS Version 1 Release 3 DFSMSrmm Guide and Reference* (SC26-7404-02).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- DFSMSrmm manages data sets and volumes in a set with functions provided in the DFSMSrmm ISPF dialog.
- DFSMSrmm records the expiration date for data sets on a volume.

Changed Information

The following information was changed in this edition:

- The DFSMSrmm RMM TSO subcommand parameters MEDIATYPE and RECORDINGFORMAT have been changed to support the IBM TotalStorage Enterprise Tape System 3592.

Summary of Changes for SC26-7404-02 z/OS Version 1 Release 3

This document contains information previously presented in *z/OS Version 1 Release 3 DFSMSrmm Guide and Reference* (SC26-7404-01).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- Information about using the MODIFY command to check journal status has been added.
- Information about using duplicate volume serial numbers has been added.
- New REXX variables: EDG@CDSU, EDG@JBDT, EDG@JBTM, EDG@JRNS, EDG@VOL1, EDG@CMDD, and EDG@CMDO.
- Users of RMM TSO subcommands need to be authorized to issue the following subcommands: CHANGEDATASET, CHANGEVOLUME, DELETEVOLUME, LISTDATASET, LISTVOLUME, LISTVRS, SEARCHDATASET, SEARCHVOLUME, and SEARCHVRS.
- Information about using the new VOL1 operand has been added to the ADDVOLUME, CHANGEVOLUME, and SEARCHVOLUME subcommands.
- Information about using the NOVOL1 operand has been added to the SEARCHVOLUME subcommand.
- Information about the STORAGECLASS and DATACLASS operands has been added to the CHANGEDATASET subcommand.
- Information about the NOSTORAGECLASS, NOMANAGEMENTCLASS, and NODATACLASS operands has been added to the CHANGEDATASET subcommand.
- Additional return codes and reason codes have been added.

Changed Information

The following information was changed in this edition:

- Consolidated messages that DFSMSrmm issues to callers of the EDGTVEXT or EDGDFHSM installation exits in a new section called Programming Interface Messages.
- Updated DFSMSrmm ISPF panels and RMM TSO subcommand output.
- Updated the ADDVOLUME, CHANGEVOLUME, and SEARCHVOLUME subcommand RECORDINGFORMAT operand with the 384TRACK value.
- Updated REXX variables EDG@DSC, EDG@DSEQ, and EDG@FILE with the value 65535.
- Updated authorization information for using the RMM TSO subcommands.

Summary of Changes for SC26-7404-01 z/OS Version 1 Release 3

This document contains information previously presented in *z/OS Version 1 Release 1 DFSMSrmm Guide and Reference* (SC26-7404-00).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- DFSMSrmm enhanced bin management provides better utilization of bin-managed storage locations.
- DFSMSrmm report generator introduces a new ISPF function to create reports.
- DFSMSrmm supports special characters in volume serial numbers.
- DFSMSrmm TSO help packaging is changed.
- DFSMSrmm reporting now supports production of an extended extract data set. The extract data set contains a new type of record which is a combination of the data set record and the volume record.
- DFSMSrmm enhancements for storage location management now include allowing storage locations to be defined as home locations.

Deleted Information

The following information was deleted from this edition:

- Previous ″Quick Reference″ chapter.

Changed Information

The following information was changed in this edition:

- DFSMSrmm pre-ACS support is enhanced so that you can control how policy information is assigned using the EDGRMMxx PARMLIB OPTION command PREACS and SMSACS operands.

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

Moved Information

The following information was moved within this edition:

- The information previously presented in the ″Getting Started″ and ″Quick Reference″ chapters has been merged into one ″Getting Started″ chapter.

Chapter 1. Getting Started with DFSMSrmm

This chapter is an introduction to the basic tasks you might need to perform using DFSMSrmm.

See *z/OS DFSMSrmm Implementation and Customization Guide* for an overview of DFSMSrmm.

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

Using the Dialog

This section explains how to use the DFSMSrmm dialog from the following sources:

- an ISMF selection menu
- an ISPF primary option panel
- the RMMISPF EXEC

Starting the Dialog

When you start your session, the DFSMSrmm primary option menu, as shown in Figure 1, 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 if you use the RMMISPF EXEC with an operand.

```
  Panel  Help
------------------------------------------------------------------------------
EDG@PRIM              REMOVABLE MEDIA MANAGER (DFSMSrmm) - z/OS V1R6
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.




5694-A01 (C) COPYRIGHT 1993,2004 IBM CORPORATION
```

Figure 1. DFSMSrmm Primary Option Menu

© Copyright IBM Corp. 1992, 2004

From an ISMF Selection Menu

You can select an option under ISMF to define a user mode to control which panel DFSMSrmm displays 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 1 on page 1. If you select end user mode, DFSMSrmm displays the DFSMSrmm User Menu as shown in Figure 2. See the *z/OS DFSMS: Using the Interactive Storage Management Facility* for additional information.

```
  Panel  Help
------------------------------------------------------------------------------
EDGP@USR                      DFSMSrmm User Menu - z/OS V1R6
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
R  REPORT     - Work with reports


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




5694-A01 (C) COPYRIGHT 1993,2004 IBM CORPORATION
```

Figure 2. 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.

From an ISPF Primary Option Menu

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 the *z/OS DFSMSrmm Implementation and Customization Guide*.

From 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 3 shows the operands you can use with the RMMISPF EXEC to bypass the DFSMSrmm primary option menu:

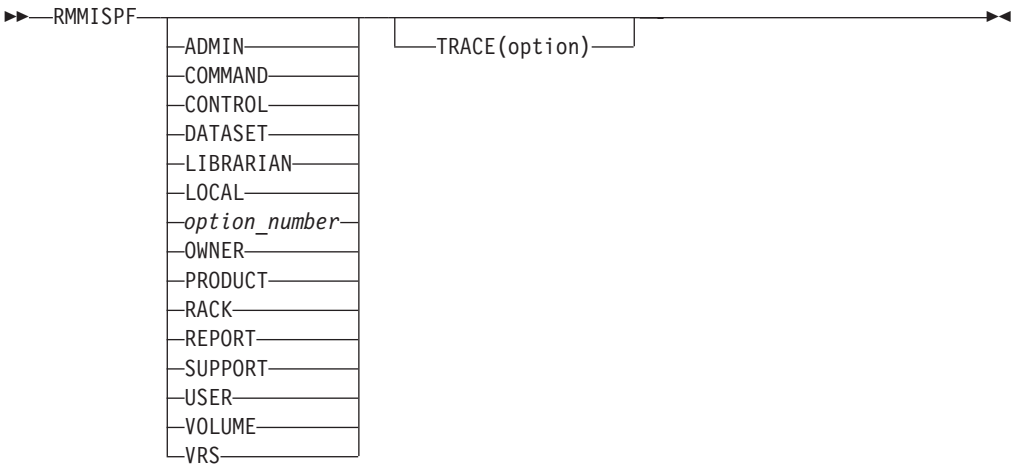


Figure 3. RMMISPF Exec Syntax Diagram

Each of the operands, except for TRACE and option_number, represents a menu from which you can request functions. For example, if you enter 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 *z/OS DFSMSrmm Diagnosis Guide*. For information on bypassing the DFSMSrmm primary option menu, see *z/OS DFSMSrmm Implementation and Customization Guide*.

Type RMMISPF and an option number from the TSO READY prompt, or type TSO RMMISPF and an option number on an ISPF Option line and press Enter. Many DFSMSrmm™ dialog options have a secondary list of options. To bypass the second menu, type two option numbers, separating them with a period. For example, entering RMMISPF 5.1 on the DFSMSrmm Primary Option Menu is the same as entering 5 on the DFSMSrmm Primary Option Menu and 1 on the DFSMSrmm Command Menu. You can continue to bypass selection menus by adding to the option number; for example, RMMISPF 5.1.5 takes you directly to the DFSMSrmm volume search panel.

User Menus

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. 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 10 for more information on the specific fast path commands for user menus.

From the primary option menu, make one of the following selections:

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

Command Menu

All users have access to 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, do one of the following tasks:

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

Figure 4 shows the Command Menu DFSMSrmm displays.

```
  Panel  Help
------------------------------------------------------------------------------
EDGP@CMD                    DFSMSrmm Command Menu - z/OS V1R6
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
R  REPORT    - Work with reports


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




5694-A01 (C) COPYRIGHT 1993,2004 IBM CORPORATION
```

Figure 4. 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 5, 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
9  STACKED   - Add one or more stacked volumes


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



5694-A01 (C) COPYRIGHT 1993,2004 IBM CORPORATION
```

Figure 5. 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.

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	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.	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	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.	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	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.	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)	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.

Changing Options

You can change DFSMSrmm processing options using the DFSMSrmm Dialog User Options panel shown in Figure 6 on page 7.

1. To change processing options, select Option 1 (USER) on the Dialog Options Menu and press ENTER. Figure 6 on page 7 shows the DFSMSrmm Dialog User Options panel that DFSMSrmm displays.

```
Panel  Help
------------------------------------------------------------------------------
EDGP@OP1                DFSMSrmm Dialog User Options
Command ===>

Date format . . . . . ISO              ( American, European, Iso or Julian )

Confirm deletes . . . YES              ( Yes or No )

Processing option . . F                 F - Foreground, B - Background

Eject option  . . . . C                 C - Convenience, B - Bulk

Variable reuse  . . . Y                 Y - Yes, N - No

Job statement information:

 ===> //*
 ===> //*
 ===> //*
 ===> //*

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

Figure 6. DFSMSrmm Dialog User Options Panel

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

- What date format should be 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
- If cartridges are ejected to a convenience input/output station or to a high-capacity output station
- If DFSMSrmm should reuse saved variable values to prime the ISPF dialog panels

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 DFSMSrmm 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 DFSMSrmm 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 twenty first century dates can be supported. For example, 15 March 99 is 15/03/1999 and 15 March 01 is 15/03/2001 in European format.

Selecting Command Processing

You can select how DFSMSrmm is to process your requests for functions. Choose one of the following options:

Foreground	DFSMSrmm processes your requests interactively
Background	DFSMSrmm saves your requests in a data set

Confirming 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 DFSMSrmm 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 DFSMSrmm 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 DFSMSrmm 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 34 for more information on the individual confirm delete panels.

Ejecting Cartridges to a Convenience Input/Output Station or a High-Capacity Output Station

You can specify if you want cartridges from system-managed tape libraries to be ejected to a convenience input/output station or a high-capacity output station.

Enter C in the Eject Option of the DFSMSrmm Dialog User Options panel if you want to eject cartridges to a convenience input/output station. Enter B if you want to eject cartridges to a high-capacity output station.

Priming DFSMSrmm ISPF Dialog Panels

You can specify if you want DFSMSrmm to prime the dialog panels with information that DFSMSrmm has saved from a previous session.

Enter Y in the Variable reuse Option of the DFSMSrmm Dialog User Options panel if you want DFSMSrmm to reuse saved information. Enter N if you do not want saved information to be reused.

Displaying Options

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

- Select Option 0 (OPTIONS) from any menu and press ENTER.
- 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 7.

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

1  USER      - Specify processing options
2  SORT      - Specify list sort options
3  REPORT    - Specify report options


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









5694-A01 (C) COPYRIGHT 1993,2004 IBM CORPORATION
```

Figure 7. DFSMSrmm Dialog Options Menu

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 the 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 123 for information on requesting functions using list line operators from lists, and Chapter 9, "Using RMM TSO Subcommands," on page 163 for information on requesting functions using the TSO command and subcommands.

Issuing Fast Path Commands

Related Reading: Use the fast path commands listed in Appendix B, "DFSMSrmm ISPF Dialog Fast Path Commands," on page 417 to display specific panels.

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 5 on page 5 for an example of this panel.)

If you type:

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

DFSMSrmm displays the DFSMSrmm Volume Display panel (Figure 8), which you use to request information about a specific volume.

```
  Panel  Help
------------------------------------------------------------------------------
EDGPT100                  DFSMSrmm Volume Display
Command ===>

Volume serial . . .









Enter the volume serial and press ENTER to display volume information.
```

Figure 8. 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.

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

You can stop your DFSMSrmm session in one of the following ways:

- **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 9 on page 12 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 DFSMSrmm 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 9. 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 perform the following steps:

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:

<tso prefix>.RMM*nn*.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.

Using Lists

This section describes the following tasks associated with lists:

- Issuing the SORT command
- Using SORT options panels
- Changing the default sort order
- Printing display lists and panels
- Requesting batch details

Issuing the SORT Command

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 117 for more information.

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, do one of the following tasks:

- Select Option 2 (SORT) on the Dialog Options Menu and press ENTER.
- Type `OPTIONS SORT` on the command line of any DFSMSrmm panel and press ENTER.

Figure 10 shows the DFSMSrmm 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.









5694-A01 (C) COPYRIGHT 1993,2004 IBM CORPORATION
```

Figure 10. 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 DFSMSrmm Data Set List Sort Options panel, as Figure 11 on page 14 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 11. 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 DFSMSrmm list sort options panels are similar, except for the table field names used in the different DFSMSrmm lists. For more information on using the DFSMSrmm sort options panels and on the various table field names, see “How to Sort a List” on page 117.

Changing the Default Sort Order

On the DFSMSrmm 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 do one of the following:

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 DFSMSrmm 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 DFSMSrmm 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 DFSMSrmm Volume List Sort Options panel, as Figure 12 on page 15 shows:
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.

```
  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 12. DFSMSrmm Volume List Sort Options Panel

Printing Display Lists and Panels

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 the *z/OS ISPF User's Guide Volume I* for more information.

Requesting Batch Details

You can request lists of resources and details about those resources in batch, by performing the following tasks:

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, perform the following tasks:

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.

Using Volumes

This section describes the following tasks associated with volumes.

For Information about	See
Requesting a scratch volume	“Requesting a Scratch Volume”
Changing volume information	“Changing Volume Information” on page 18
Changing the release date for a volume	“Changing the Release Date for a Volume” on page 18
Setting release actions for a volume	“Setting Release Actions for a Volume” on page 19
Changing data set information	“Setting Release Actions for a Volume” on page 19
Changing owner information	“Changing Owner Information” on page 19
Releasing volumes manually	“Releasing Volumes Manually” on page 19
Requesting notification of release	“Requesting Notification of Release” on page 20
Requesting return after release	“Requesting Return after Release” on page 20
Retaining data sets and volumes	“Retaining Data Sets and Volumes” on page 20
Grouping volumes	“Grouping Volumes” on page 21

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 EDGRMM*xx* MASTEROVERWRITE operand. We recommend that you request scratch volumes in this way.

To manually request a scratch volume without running a batch job, follow these steps:

1. Select Option 5 (REQUEST) on the User Menu and press ENTER.

   DFSMSrmm displays the 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.

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 234 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.
2. Enter any of the following information 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.

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

   DFSMSrmm returns a list of volumes in the DFSMSrmm 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 DFSMSrmm Change Volume Details panel, containing information about the volume you specified.
7. Change information or add missing information to any of the fields on the panel, and press ENTER. You cannot change VOL1, RACK, or the PREVIOUS VOLUME fields.

Changing the 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. 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 1999/195.

If you want the volume to be considered for release immediately, you can release it manually. See “Releasing Volumes Manually” on page 19 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 DFSMSrmm 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 Owner Information

To change information in your owner record, follow these steps:

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 the DFSMSrmm Change Owner Details panel that defines information for your owner ID.
3. Change information on the panel or add any information missing from your owner information. For example, to be notified when a volume you own is about to be released, include a user ID and node in your owner information. 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.

Releasing Volumes 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 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 to confirm release requests before DFSMSrmm releases the volumes, DFSMSrmm displays a DFSMSrmm Confirm Volume Release panel. Press ENTER to confirm that you want to release the volume. Press PF3 to cancel the request.
   - If you requested a list, DFSMSrmm displays a list of volumes you own. Type R in the line operator column (S) beside the volume serial number for each volume you want to release and press ENTER.

Requesting Notification of Release

If you want to be notified when a volume you own is being considered for release, use the DFSMSrmm 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 19 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 DFSMSrmm Change Owner Details panel to add missing information or to change an incorrect user ID and node. See “Changing Owner Information” on page 19 for more information.

Requesting Return 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 DFSMSrmm Change Volume panel to do this anytime before the volume is due to be released. See “Setting Release Actions for a Volume” on page 19 for more information.

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 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 that the pool must accommodate, and the number of volumes to be added in the future.
- 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?

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 219 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.

   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.

   Specify a volume serial number and specify Yes for List entire set and DFSMSrmm displays all the data sets in a multi-volume set.

Press ENTER to request a list of data sets on volumes you own. DFSMSrmm returns a list of data sets. Press PF11 to see additional data set information. Find the data set you want to change and type C in the line operator column (S), beside the data set name. Press ENTER.

DFSMSrmm displays a DFSMSrmm Change Data Set Details panel containing information defined for the data set.

2. Enter information in the DFSMSrmm Change Data Set Details 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.

3. Press ENTER to request your changes.

Chapter 2. Defining Your Resources Using the DFSMSrmm ISPF Dialog

Related Reading

Throughout this topic, you will find references to help panels for additional information and to the RMM TSO subcommands you can use instead of panels. See Chapter 9, “Using RMM TSO Subcommands,” on page 163 for more information on using TSO subcommands.

This topic 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 the volume. DFSMSrmm records this information each time a data set is opened and closed.

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 are optional and define individual shelf locations in your removable media library. Rack numbers are six alphanumeric, national, or special 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 DFSMSrmm depending on your installation definition.

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.

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.

© Copyright IBM Corp. 1992, 2004

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 182 or “ADDBIN: Adding a Bin Number in a Storage Location” on page 166 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 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, 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 272 or “DELETEBIN: Deleting Bin Number Information” on page 263 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.
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 the number of 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 the number of 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

You can define physical, logical, and stacked volumes to DFSMSrmm. See the *z/OS DFSMSrmm Implementation and Customization Guide* for information about defining stacked volumes to DFSMSrmm.

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 the *z/OS DFSMSrmm Implementation and Customization Guide* for information.

DFSMSrmm obtains information for volumes residing in a system-managed library from the tape configuration database (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 186 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 the criteria set by the DFSMSrmm EDGRMM*xx* OPTION 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 or expired data on them or because the data is no longer needed.

When you manually define volumes, you can optionally 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, and optionally a system managed library name. Your installation defines the pools to which you can associate volumes. If you use a rack number or pool ID of the shelf location where the volume is to reside, the rack number or pool ID must be already defined to DFSMSrmm before you can define the volume. When you do not define a shelf location in one of these ways, and the volume is added in location SHELF, DFSMSrmm assumes that the shelf location name matches the volume serial number.

When you are adding volumes to a manual tape library, you must ensure that a valid media type is defined to DFSMSrmm. If the media type is incorrect, the system cannot allocate the correct tape drives.

Specifying Media Type and Recording Format for Volumes

You can specify the media name and recording format for volumes. Table 5 show the correspondence between the DFSMSrmm media type names and the media names used in the system-managed tape data class construct.

Table 5. DFSMSrmm Media Types

The Media Type	Corresponds to the DFSMSrmm Media Name
MEDIA1	CST
MEDIA2	ECCST
MEDIA3	HPCT
MEDIA4	EHPCT

Table 5. DFSMSrmm Media Types (continued)

The Media Type	Corresponds to the DFSMSrmm Media Name
MEDIA5	ETC
MEDIA6	EWTC
MEDIA7	EETC
MEDIA8	EEWTC

Table 6 lists the current media types and recording formats. The first column shows the media types that you can specify and the default value. The second column shows valid recording formats for each media type. The third column shows the default media type set by DFSMSrmm when you do not specify a value or when you specify a value that is not valid for the specified media type.

As shown in Table 6, if you specify media types CST or ECCST, you can only specify 18TRACK and 36TRACK recording formats. If you specify media types HPCT or EHPCT, you can only specify 128TRACK, 256TRACK, or 384TRACK recording formats. If you specify media types ETC/MEDIA5, EWTC/MEDIA6, EETC/MEDIA7, or EEWTC/MEDIA8, you can only specify EFMT1 recording format.

Table 6. How DFSMSrmm Assigns Media Type and Recording Format

Media type	Recording format	Default recording format
Not Specified/*	* 18TRACK 36TRACK 128TRACK 256TRACK 384TRACK EMFT1	*
CST	18TRACK 36TRACK	18TRACK
ECCST	18TRACK 36TRACK	36TRACK
HPCT	128TRACK 256TRACK 384TRACK	128TRACK
EHPCT	128TRACK 256TRACK 384TRACK	128TRACK
ETC/MEDIA5	EFMT1	EFMT1
EWTC/MEDIA6	EFMT1	EFMT1
EETC/MEDIA7	EFMT1	EFMT1
EEWTC/MEDIA8	EFMT1	EFMT1

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 186 for more information about the ADDVOLUME subcommand.

Follow these steps 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.

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 RMM ADDVOLUME subcommand to add information about scratch volumes to DFSMSrmm. See “ADDVOLUME: Adding Volume Information” on page 186 for more information about the ADDVOLUME subcommand.

Follow these steps 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.
2. Supply a volume serial number.

   You must specify COUNT value 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. You can, for example, specify either a pool ID or a rack number to direct where the scratch volumes should be stored in the removable media library. If you choose to specify a pool ID or a rack number, 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.

Adding Volumes with Duplicate Volume Serial Numbers

You can add a duplicate volume using the DFSMSrmm ISPF dialog or the RMM ADDVOLUME subcommand. A duplicate volume is defined to DFSMSrmm with a unique external volume serial number and a VOL1 label that might duplicate another volume but that does not match its own external volume serial number.

For example, you have defined a pool of volumes that you use to manage duplicate volumes. You want to define a volume with the volume serial number A00001. You already have a volume with volume serial number A00001 defined to DFSMSrmm and the next available rack number in the pool is D00010 which comes from the pool that you specified for duplicate volumes.

When you use the DFSMSrmm ISPF dialog to add a volume that already exists, DFSMSrmm displays the Add Duplicate Volume panel. Follow these steps to add a volume with a duplicate volume serial number.

1. Select the option to add the volume as a duplicate or to change the existing volume to be a duplicate volume.
2. Specify a pool for the duplicate volume.
3. Press enter.

Redefining a Volume Already Defined to DFSMSrmm

If you chose not to use duplicate volume serial numbers, you can either redefine the old volume before you add the new volume or use a different volume serial number to add the new volume. To redefine the volume or to use a different volume serial number for the volume, follow these steps:

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 RMM DELETEVOLUME FORCE subcommand. You cannot delete a volume using the DFSMSrmm ISPF dialog. If you use RMM DELETEVOLUME without specifying the FORCE operand, DFSMSrmm sets the volume status to pending release.
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. See the *z/OS DFSMSrmm Implementation and Customization Guide* for information on using the EDGUX100 installation exit to ignore known volumes or to manage volumes with duplicate volume serial numbers.

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 234 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 234 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, follow these steps.

1. Select Option 3 (CHANGE) on the Volume Menu and press ENTER.
2. Enter the serial number of the volume for which you want to change information.
3. Press ENTER.
4. Make changes or add missing information to any of the fields on the Change Volume Details panel. You can scroll down to view additional fields and change volume information. Press PF1 for field-specific help information.
5. Press ENTER to process your changes.

Once you have defined a stacked volume to DFSMSrmm, you do not normally need to change information about that volume. DFSMSrmm manages the movement of stacked volumes once they are exported from the VTS library.

You can use DFSMSrmm commands to perform a virtual export for a logical volume to an existing exported stacked volume container under these conditions.

- You have imported a logical volume from a stacked volume.
- You have processed the logical volume only for input.
- You want to re-associate the logical volume with the previously exported volumes on the original stacked volume container.

Use the RMM CHANGEVOLUME subcommand with the CONTAINER operand to perform the virtual export by issuing a command like the one that follows.

```
RMM CHANGEVOLUME V12345 CONTAINER(S26901)
```

Changing Information for Volumes in Multivolume Sets

You can change information for volumes in multivolume sets by using the DFSMSrmm ISPF dialog.

1. Select Option 3 (CHANGE) on the Volume Menu and press ENTER.
2. Enter a volume in a multivolume set in the Change Volume panel and press Enter. DFSMSrmm displays the panel Figure 13. Enter Y to update all volumes in a multi-volume set.

```
EDGPT330
  Do you want to process all volumes in the multi-volume set ?  _ (Y/N)

Enter=Execute
```

Figure 13. DFSMSrmm Processing Multiple Volumes

3. Change information or add missing information to any of the fields on the Change Volume Set panel. You cannot change the RACK, VOL1, and PREVIOUS VOLUME information when you change the volume information in a multivolume set. Press ENTER to process your changes. Press PF1 for field-specific help information.
4. DFSMSrmm displays the DFSMSrmm Processing Volume Set panel which contains a list of all the volumes in the multivolume set that are to be updated. Press Enter to update all the volumes in the multivolume set. DFSMSrmm displays the results of processing in the Change Processed column. The values for the Change Processed column can be OK if the volume is successfully processed or ERROR with a return code and reason code if an error occurred during processing.

Deleting Volume Information

DFSMSrmm deletes volume information from the control data set when you release a volume. See “Releasing Volumes” on page 102 for more information.

You can delete stacked volumes from DFSMSrmm only when they contain no volumes. When stacked volume support is enabled, DFSMSrmm tracks the number of contained volumes. To remove a volume from a stacked volume, use the RMM CHANGEVOLUME subcommand on the contained volume to clear the container name.

Deleting Information for Volumes in Multivolume Sets

You can delete information for volumes in multivolume sets by using the DFSMSrmm ISPF dialog.

1. Select Option 4 (RELEASE) on the Volume Menu and press ENTER.
2. Enter a volume in the Release Volume panel (EDGPT400) and press ENTER. If the volume is a volume in a multivolume set, DFSMSrmm displays Figure 13 on page 30. Enter Y to delete or release all volumes in a multi-volume set.
3. DFSMSrmm displays the DFSMSrmm Processing Volume Set panel which contains a list of all the volumes in the multi-volume set that are to be processed. Press Enter to delete or release all the volumes in the multivolume set. DFSMSrmm displays the results of processing in the Change Processed column. The values for the Change Process column can be OK if the volume is successfully processed or a return code and reason code if an error occurred during processing.

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 the user. DFSMSrmm 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 176 for more information on the ADDOWNER subcommand.

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

   DFSMSrmm displays the 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 229 for more information.

Restriction: 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.
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 268 for more information.

Restriction: 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. Press ENTER.

   If you did not request the confirm delete option and the owner ID you specified still owns volumes, DFSMSrmm displays the 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 on the DFSMSrmm Delete Owner panel. 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 do not 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 179 for more information on the ADDPRODUCT subcommand.

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.
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.
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 product volumes.

1. Enter the 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. Press PF1 for field-specific help information. See *z/OS DFSMSrmm Implementation and Customization Guide* for information on how to change these values if necessary.

   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 25 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 232 for more information.

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.

To change information recorded by DFSMSrmm for a 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.
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. 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, 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 123 for more information.
4. Press ENTER to process your changes.

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 270 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.

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.

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
- Expiration date which is the date that the data set can be considered for deletion.
- 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 ADDDATASET subcommand to add information about a data set to DFSMSrmm. See “ADDDATASET: Adding Data Set Information” on page 169 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.
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 219 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.
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 for the data set you defined.
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 266 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.

   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.

Restriction: 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.

When you request the confirm delete option for a session, DFSMSrmm displays the 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.

Press ENTER to delete information about the data set you specified and all subsequent data sets on the volume. If you specified the DFSMSrmm parmlib member OPTION UNCATALOG(Y) operand, DFSMSrmm also uncatalogs the data set and all subsequent data sets on the volume for which DFSMSrmm has recorded information.

Chapter 3. 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. Use DFSMSrmm inventory management to process the retention and movement policies specified in vital record specifications.

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.

You can control the way DFSMSrmm retains and moves data sets and volumes by defining these DFSMSrmm parmlib member EDGRMM*xx* OPTION operands.

- Use the DFSMSrmm parmlib OPTION VRSCHANGE operand to force running trial run vital record processing when you have made changes to vital record specifications. You must allocate an ACTIVITY file when you run trial run processing. The ACTIVITY file 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. See the *z/OS DFSMSrmm Reporting* for a sample you can use to create the ACTIVITY report.
- Use the DFSMSrmm parmlib OPTION VRSMIN operand to set the minimum number of vital record specifications that you must define for you to run inventory management vital record processing. See “Defining the Minimum Required Policy” on page 94 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 86 for more information about selecting vital record processing.
- Use the DFSMSrmm OPTION RETAINBY operand or the OPTION MOVEBY operand to retain volumes or move volumes as sets or individually. See “Retaining and Moving Volumes as Sets or Individually” on page 61.

See the *z/OS DFSMSrmm Implementation and Customization Guide* for information about 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.

© Copyright IBM Corp. 1992, 2004

DFSMSrmm applies retention and movement policies to data sets by matching a vital record specification to a data set by 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 41 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. The name vital record specification 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 to define move information only. Specify VRSEL(NEW) 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 50 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 that are 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 14 and Figure 15 on page 41 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 14 points to N1.

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

Figure 14. Chaining Vital Record Specifications

Figure 14 on page 40 uses the name vital record specification N1 shown in Figure 15 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 15. 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 16, a vital record specification is defined to retain a maximum of 20 cycles for all data set matching to the mask 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. DFSMSrmm retains extra cycles 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 16. 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 91 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 by 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 that are 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 43 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 by 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 that match a 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 48 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 *z/OS 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 by 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 that are based on the value you set for the DFSMSrmm EDGRMM*xx* parmlib member OPTION VRSEL operand.

Table 7 on page 44 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 7. How DFSMSrmm Matches Data Set Names to Data Set Masks

Order (OLD)	(NEW)	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 7 shows the order that DFSMSrmm uses to match data sets that are defined to DFSMSrmm to data set name masks in vital record specifications when VRSEL(NEW) is specified. If DFSMSrmm matches a data set to 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.

When VRSEL(NEW) is specified, vital record groups not only have the same vital record specifications, the same data set names, and optionally, the same job names, but also the same retention types. Also, when a data set matches both a primary and secondary vital record specification, the management value or class retention type is used in determining the correct vital record group.

Also, when VRSEL(NEW) is specified, grouping is important when data sets are retained by a cycles retention type of either CYCLES or BYDAYSCYCLE. A cycles retention type can be specified in any retention vital record specification in either the primary or the secondary vital record specification, or both. To calculate grouping, DFSMSrmm considers both the primary and the secondary vital record specifications. If any data sets match either a management class or a management value vital record specification, the grouping is determined twice. Once using the management class and management value vital record specifications, and once using the matching data set name mask vital record specification. This happens because the job name and group data set name may be different between the two and also because the grouping for management class and management value vital record specifications with a cycles retention type must ensure that all data sets of the same name that match to these vital record specifications are processed as a group to allow cycles to be determined. The groups are determined as follows:

- For data sets matching a management class or a management value vital record specification, the group is determined using the data set name, optionally the job name, and the matching management class or management value vital record specification. If the first vital record specification in the chain specifies a cycles retention type, the matching vital record specifications are considered equal.
- For data sets matching a vital record specification using the data set name mask, the group is determined using the group data set name, optionally the job name, and the matching data set name vital record specification. During processing, the retention decision taken in the first group is used when an UNTILEXPIRED retention type is found.

DFSMSrmm processes each data set in a group, applying the policies from the matching value vital record specification to determine if the data set is retained as part of the group. A data set is retained by vital record specification if either or both of the group processing identifies the data set for retention.

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 that are based on the job that created the data set. Specify a job name by 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 names 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 17 shows examples of RMM ADDVRS subcommands that use 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 17. Valid Job Name Masks

How DFSMSrmm Uses Jobname to Apply Policies

Table 8 on page 47 shows how DFSMSrmm applies policies based on the jobname you specify and the DFSMSrmm parmlib EDGRMMxx member OPTION VRSJOBNAME operand value you use.

Table 8. 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 9 shows the order that DFSMSrmm uses to match job names defined to DFSMSrmm to job name masks in vital record specifications.

Table 9. 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 that are created by JOB1; data sets that are created by job JOB2; data sets that are 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 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 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 18 defines a retention policy for a GDG base name, SOTIRI.RETAIN.

```
RMM ADDVRS DSNAME('SOTIRI.RETAIN') GDG CYCLES COUNT(3)
```

Figure 18. 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 19 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 19. 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 the following example.

  ```
  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 the following example.

  ```
  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.

DB2® systems and DFSMShsm systems 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 Figure 20, 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)
```

Figure 20. Specifying Data Set Name Masks to Manage Data Sets Like a GDG

To define a vital record specification for all GDGs not covered by any other vital record specification, you could use the data set name mask shown in Figure 21.

```
RMM ADDVRS DSNAME('**.G¬¬¬¬Y¬¬') NOGDG
```

Figure 21. Specifying A Global Data Set Name Mask to Manage GDGs

Restriction: 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 22 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 22. Specifying NOGDG and CYCLES in a Vital Record Specification

Figure 23 shows an example of defining two vital record specifications for the same data set name by 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¬¬¬¬V¬¬) -
      NOGDG DAYS COUNT(5)
```

Figure 23. 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. In DFSMSrmm volume serial numbers are used to identify volumes and to identify the volume label. DFSMSrmm allows you to define a volume by using a different serial number than the one that is recorded in the volume label. In this way you can define volumes with duplicate volume serial numbers.

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 that begin 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 Using the DFSMSrmm ISPF Dialog

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 205 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.
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

To add a data set vital record specification, follow these steps.

1. Enter 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 and then displays the Add Data Set VRS panel.
2. Enter any information you want DFSMSrmm to record for the vital record specification. All fields on this panel are optional. Press PF1 for help.
3. Press ENTER to add the vital record specification.

Adding Volume Vital Record Specifications

To add a volume vital record specification, follow these steps.

1. Enter 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. DFSMSrmm then displays the Add Volume VRS panel.
2. Enter any information you want DFSMSrmm to record for the vital record specification. All fields on this panel are optional. Press PF1 for help.
3. Press ENTER to add the vital record specification.

Adding Name Vital Record Specifications

To add a name vital record specification, follow these steps.

1. Enter a name on the Add Vital Record Specification panel. DFSMSrmm checks to see if a vital record specification with the same name already exists. DFSMSrmm then displays the Add Name VRS panel.
2. Enter any information you want DFSMSrmm to record for the vital record specification. All fields on this panel are optional. Press PF1 for help.
3. Press ENTER to add the vital record specification.

Changing Vital Record Specifications

Related TSO Subcommand

To change information about a vital record specification by using RMM 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 307, “DELETEVRS: Deleting Vital Record Specifications” on page 279, and “ADDVRS: Adding a Vital Record Specification” on page 205 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.
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.

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 RMM DELETEVRS subcommand to delete a vital record specification defined to DFSMSrmm. See “DELETEVRS: Deleting Vital Record Specifications” on page 279 for more information.

When a vital record specification is deleted, DFSMSrmm does not change any data set or volume information. The changes take place during the next vital record processing run. DFSMSrmm uses only the remaining vital record specifications to apply policies. If a data set or a volume matches to another remaining vital record specification, DFSMSrmm applies those policies defined in the vital record specification. If the data set or volume does not match to any vital record specifications, then the data set or volume is no longer retained by a vital record specification and 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 .
2. Enter a data set name, a volume serial number, or a vital record specification name. You must specify a data set name before you can specify a job name.
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 linked to it.
4. Press ENTER to delete the vital record specification.

Chapter 4. Retaining and Moving Your Volumes

Use vital record specifications to define retention and movement policies for data sets and volumes. See Chapter 3, “Defining Vital Record Specifications,” on page 39 for information about defining data set vital record specifications 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 58 and “Defining Movement Policies for Volumes” on page 70 for more information. See the*z/OS DFSMSrmm Implementation and Customization Guide* for information about DFSMSrmm support for stacked volumes.

DFSMSrmm identifies the volumes to 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, 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. See “Global Confirmation” on page 82 for examples of using the RMM CHANGEVOLUME subcommand to perform global confirmation.

DFSMSrmm creates a detailed report of data sets and volumes that are retained, the vital record specification used, and the location where the data set or volume is retained. See the *z/OS DFSMSrmm Implementation and Customization Guide* for more information about running DFSMSrmm inventory management. See the *z/OS DFSMSrmm Reporting* document for information about creating inventory management reports. DFSMSrmm keeps the latest retention information for each data set in the control data set. Use the RMM LISTDATASET subcommand described in “LISTDATASET: Displaying Information about a Data Set” on page 295 to request retention information.

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 with the minimum number of cycles or copies to be kept for a matching data set. 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)
```

© Copyright IBM Corp. 1992, 2004

BYDAYSCYCLE

Define a vital record specification with the minimum number of cycles or copies that should be kept for a matching data set. For BYDAYSCYCLE processing, DFSMSrmm considers all the data sets created in a single day to be a cycle.

Example

Define a vital record specification with the minimum number of cycles or copies to be kept for a matching data set. 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. DFSMSrmm retains all data sets matching the data set name mask created during this period.

Example

To retain data sets that are created in the last 30 days with the high-level qualifier 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. The DFSMSrmm EDGRMMxx parmlib OPTION VRSEL(NEW) operand 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 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 *z/OS DFSMSrmm Implementation and Customization Guide* for information about specifying the parmlib OPTION UNCATALOG operand to control how DFSMSrmm uncatalogs data sets.

Data sets that are created by long-running batch jobs might become opened before but become cataloged after a run of inventory management. Use the parmlib OPTION CATRETPD to protect any of these data sets that match a vital record specification that specifies WHILECATALOG and to retain them in their current location. The CATRETPD option specifies the number of hours DFSMSrmm retains catalog-controlled data sets even if they have never been cataloged. DFSMSrmm

retains the data set for the catalog retention period if the data set has never been cataloged. DFSMSrmm does not retain the data set if the data set has been cataloged and uncataloged during the catalog retention period. Setting the parmlib OPTION CATRETPD(12) RETPD(0) MAXRETPD(0) ensures that DFSMSrmm vital record specifications only controls retention.

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, 1999 is higher than an expiration date of September 1, 1998.

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 retain the data set 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. Define vital record specifications to manage these data sets with 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 there is no creating job name 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. All data sets and volumes that match the vital record specification become eligible for release processing. Be aware that once the vital record specification is deleted, the data set can match to another less specific vital record specification and might be retained by the less specific vital record specification.

Example

To retain the data set, 'MIKES.DONT.DELETE.UNTIL.BTHDAY', until either the deletion date 2001/361 is reached or the data set is uncataloged, issue the command:

```
RMM ADDVRS DSNAME('MIKES.DONT.DELETE.UNTIL.BTHDAY') COUNT(99999) -
   DELETEDATE(2001/361) LOCATION(REMOTE) STORENUMBER(99999) -
   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 in a vital record specification overrides the expiration date set for a data set or volume and can extend or reduce 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. Use DFSMSrmm installation exit EDGUX100 to 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. 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 by 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 by using SMS management class names, or by using 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 that use 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 by 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, define a vital record specification management value. Use the DFSMSrmm-supplied EDGUX100 installation exit as described in the *z/OS DFSMSrmm Implementation and Customization Guide* to assign management values to data sets.

Figure 24 defines a policy by 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 24. Retaining a Data Set Using Management Class

Figure 25 defines a policy by 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 25. 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, 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. 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 Figure 26, the data set is retained until the date 2001/361 when the vital record specification is deleted, the data set is uncataloged, and the data set has not been referenced for 100 days.

```
RMM ADDVRS DSNAME('RTEAM.HAS.FUN') COUNT(100) -
 DELETEDATE(2001/361) LOCATION(REMOTE) WHILECATALOG
```

Figure 26. Specifying Combinations of Retention Policies on a Single Vital Record Specification

You can request that a data set be 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 27 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 27. 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 use 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. However, there are exceptions: when a data set matches to a data set name vital record specification with the UNTILEXPIRED operand or a management value vital record specification with the WHILECATALOG operand, as shown in Figure 28. When this match occurs, only one of the two conditions must be true to retain a volume.

```
RMM ADDVRS DSNAME('A.**') UNTILEXPIRED
RMM ADDVRS DSNAME('D99000') WHILECATALOG
```

Figure 28. Merging UNTILEXPIRED with the WHILECATALOG Retention Policy

Retaining and Moving Volumes as Sets or Individually

Use the DFSMSrmm parmlib member EDGRMMxx OPTION RETAINBY or MOVEBY operands to retain or move volumes as a set or as individual volumes. The default setting is for DFSMSrmm to retain or move volumes as individual

volumes. See the *z/OS DFSMSrmm Implementation and Customization Guide* for information about the DFSMSrmm parmlib member.

Defining Release Policies

You can include release options in a data set vital record specification to control data set retention and immediate return to scratch status for volumes. You enable this processing using the DFSMSrmm EDGRMMxx parmlib OPTION VRSEL(NEW) operand.

When a data set is created on tape, DFSMSrmm uses the default retention period to calculate a volume expiration date. Use vital record specifications retention policies to extend data set retention, and therefore volume retention. Use the ADDVRS RELEASE(EXPIRYDATEIGNORE) operand to release a volume even if the volume expiration date has not yet been reached.

When a volume is released and is pending release status, you must run inventory management a second time to return the volume to scratch status. Use the ADDVRS RELEASE(SCRATCHIMMEDIATE) operand to return a volume to scratch status in a single run of inventory management when catalogs and TCDBs are shared.

These release policies apply to a volume if any of the data sets on the volume are retained by a vital record specification, and even if one of the data sets was never retained by a vital record specification or retained because of a job control language specified retention period or because of a JCL-specified retention period. Use data set name masks and jobname masks to select release policies. You can also implement an installation-wide policy that applies to all data sets retained by vital record specifications which ignores user-specified retention. If you specify the UNTILEXPIRED operand in the vital record specification, you can honor the user-specified retention.

Each time that inventory management vital record processing is run, DFSMSrmm matches the vital record specifications to the data sets to determine which release options are in effect. If a vital record specification retains any data set on a volume and a release option is specified, DFSMSrmm sets the volume release options. If, on a subsequent run of inventory management vital record processing, data sets on a volume are retained by a vital record specification and no release options are defined, DFSMSrmm resets the volume release options. For volumes with multiple data sets, DFSMSrmm uses the release option applicable during the last run of vital record processing that retained the data set on that volume.

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 date 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 3, “Defining Vital Record Specifications,” on page 39 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 primary vital record specification. Use the UNTILEXPIRED operand in a primary vital record specification 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 by using the primary vital record specification.

When both primary and secondary vital record specifications match the data set name, and you have specified UNTILEXPIRED in a primary vital record specification, you are asking DFSMSrmm to combine the retention information in both vital record specifications. 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 10 on page 64 or with DFSMSrmm special date formats.

Table 10. 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/*ccccc* 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. 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 using 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 11 on page 66 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 retention date for the data set by using the current vital record subchain as follows.

1. CATRETPD, when the WHILECATALOG retention type retains CATRETPD, but is not yet cataloged.
2. The earliest actual date. The actual date is the day on which the current matching vital record specification no longer retains the data set. If a data set is permanently retained, the retention date is set to PERMANENT and not 1999/365. DFSMSrmm sets the retention date based on one of these dates.
   a. The date a data set is created or last referenced when the DAYS or LASTREFERENCEDAYS retention type is specified.
   b. 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 a DELETEDATE other than DELETEDATE(1999/365) is specified.
3. WHILECATLG, if WHILECATALOG is specified.
4. 'CYCL/ccccc', if BYDAYSCYCLE or CYCLES is specified.
5. PERMANENT, which specifies that the data set is retained forever.

Note: DFSMSrmm uses the retention date that is the earliest when a data set matches both a primary and a 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. PERMANENT
2. CYCL/*ccccc*
3. WHILECATLG

4. A date in the format selected by your installation
5. CATRETPD

Table 11 and Table 12 on page 67 describe how DFSMSrmm calculates the retention date.

Table 11. 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/ccccc
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
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.

Table 11. 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:

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. CATRETPD is used when the data set is retained by the catalog retention value and the WHILECATALOG retention type is specified. DFSMSrmm does not retain the data set if DFSMSrmm detected that the data set was cataloged and then uncataloged during the catalog retention period.
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 12 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 12. DFSMSrmm Retention Date Calculation by COUNT(99999)

If you have Retention Type:	RMM Calculates Retention Date as:	And Displays Retention Date as:
CYCLES	vital record specification delete date OR cycles date format[1]	Date format specified by your installation or CYCL/nnnnn
DAYS	vital record specification delete date	Date format specified by your installation or PERMANENT if the VRS delete date is 1999/365
EXTRADAYS (XD)	vital record specification delete date	Date format specified by your installation or PERMANENT if the VRS delete date is 1999/365
LASTREF	vital record specification delete date	Date format specified by your installation or PERMANENT if the VRS delete date is 1999/365
CYCLES + WC	vital record specification delete date OR catalog date format[2]	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
DAYS + WC	vital record specification delete date OR catalog date format[2]	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
LASTREF + WC	vital record specification delete date OR catalog date format[2]	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.

Table 12. DFSMSrmm Retention Date Calculation by COUNT(99999) (continued)

If you have Retention Type:	RMM Calculates Retention Date as:	And Displays Retention Date as:
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 vital record specification 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 vital record specification 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 vital record specification 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 12. 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. 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. If the data set is not cataloged and CATRETPD retains the data set, DFSMSrmm uses the CATRETPD retention date. CATRETPD is used when the data set is retained by the catalog retention value and the WHILECATALOG retention type is specified. DFSMSrmm does not retain the data set if DFSMSrmm detected that the data set was cataloged and then uncataloged during the catalog retention period.
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 29 on page 70 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 29. Managing Closed-Cycle GDGs

Converting from Closed-Cycle GDGs to Using Scratch Pools

Recommendation: With closed-cycle GDGs, you might use more volumes than you actually require. 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 29. 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 30.

```
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 30. Managing Volumes Using Scratch Pools

During the first inventory management run after these changes have been made, 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 74 for more information.

Moving Physical Volumes 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 234 for more information on CHANGEVOLUME.

When you use library names to identify automated or manual tape libraries in the removable media library, DFSMSrmm validates these names by ensuring that they have been defined in the TCDB. You cannot specify a distributed library name.

See the *z/OS DFSMSrmm Implementation and Customization Guide* for information about moving logical volumes.

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 under these conditions.

- You find that the libraries are too full to accept new volumes.
- You have no more available scratch volumes.
- 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 that is 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 31 on page 72 shows how DFSMSrmm can be used to help you manage your library when it is nearing its capacity.

```
RMM ADDVRS DSNAME('**') LASTREFERENCEDAYS COUNT(99999) WHILECATALOG -
    LOCATION(HOME) STORENUMBER(10) NEXTVRS(L1)
RMM ADDVRS NAME(L1) LOCATION(LOCAL) STORENUMBER(99999)
```

Figure 31. Moving Volumes out of a System-Managed Library

The vital record specifications shown in Figure 31 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 31, LOCAL is identified as the on-site location to store volumes that have to be removed from the automated tape library. Any other storage location or a manual tape library 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 31, 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 31 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 *z/OS DFSMSrmm Implementation and Customization Guide* for more information.

Rotating Volumes In and Out of the System-Managed Library

Figure 32 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.

```
RMM ADDVRS DSNAME('USERA.MONTHLY.DATA') LASTREFERENCEDAYS COUNT(365) -
    LOCATION(LOCAL) STORENUMBER(27) NEXTVRS(N1)
RMM ADDVRS NAME(N1) LOCATION(HOME) STORENUMBER(30)
```

Figure 32. Moving a Volume in and out of the System-Managed Library

In Figure 32 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 33 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 33. Moving Data out of the System-managed Library at Fixed Intervals

In Figure 33, DFSMShsm 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 DFSMShsm.

Moving to Storage Locations

During inventory management, DFSMSrmm identifies when volumes must be moved to one or more storage locations that are 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 the RMM CHANGEVOLUME subcommand as shown in Figure 34.

```
RMM CHANGEVOLUME volser MANUALMOVE LOCATION(HOME)
```

Figure 34. 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 subcommand with the LOCATION operand. The operand LOCATION(HOME) is specified in Figure 34 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 by 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 35. Sending a Volume to Another System

The MANUALMOVE operand shown in Figure 35 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 35 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 36.

```
RMM CHANGEVOLUME volser LOCATION CONFIRMMOVE AUTOMOVE
```

Figure 36. Returning a Volume from Another System

Manually Requesting Volume Moves

To manually request a volume move, define a new location name for the volume by 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, 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 79 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 that reside . To get this information, use the RMM TSO SEARCH subcommands to list volumes that are based on loan location information. See Chapter 7, "Requesting Information about Your Resources," on page 111 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.
2. Enter a value in the Destination 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, 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 234 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, enter a library name as the new location on the Change Volume panel or on the RMM CHANGEVOLUME subcommand. Changing the location starts a movement for the volume by setting the volume destination. 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:

```
Destination ===> HOME
```

When you give the name of a system-managed library as a new location name for a volume, DFSMSrmm first checks if the volume serial number and library name are 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, the operator must move the volume to the automated tape library. During cartridge entry processing to an automated tape library, DFSMSrmm checks:
  - That the volume destination matches the entered library or is not set
  - That the volume does not duplicate a volume in another library
  - That the volume can be used on MVS systems
  - Whether the volume is owned by another library partition
  - That the rack number, if any, and the volume serial number match

  DFSMSrmm automatically confirms the volume move as complete. If DFSMSrmm was inactive during cartridge entry processing, and you allowed the entry processing, you must manually confirm that the volume was moved.
- For a move to a manual tape library, 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 defining the volume to DFSMSrmm, DFSMSrmm adds the volume information to 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 is already defined for the volume, DFSMSrmm corrects the library name in the control data set, but does not change the volumes' home location. 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 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, except for a logical volume residing in a VTS, the volume is physically ejected and must be moved to its destination by the operator. 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, 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, 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 37 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 37. 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:

```
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 libraries.

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 that is 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 that is 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. If a volume is stored in a system-managed tape library, DFSMSrmm automatically ejects the volume if you define a loan location.

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. A loan location value can be 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 38 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 38. Moving a Volume to a Loan Location

Figure 39 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 39. 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 40 to clear the loan location value.

```
RMM CHANGEVOLUME volser LOANLOC('')
```

Figure 40. Returning a Volume from a Loan Location

See “CHANGEVOLUME: Changing Volume Information” on page 234 and “SEARCHVOLUME: Creating a List of Volumes” on page 329 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.

Exception: 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.
- 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,

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, enter the volume into the automated tape library, 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 41 on page 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 41. DFSMSrmm Volume Action Status Panel

3. Type `C` next to the type of move you want to confirm as shown in Figure 41. 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 moving 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 42 on page 82:

```
  Panel  Help
------------------------------------------------------------------------------
EDGPT610            DFSMSrmm Volume Action Summary List       Row 1 to 4 of 4
Command ===>                                                 Scroll ===> PAGE

Enter HELP or PF1 for the list of available line commands.
   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 42. DFSMSrmm Volume Action Summary List

3. Use the C line operator or the CM line operator 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 that the volume has no outstanding release actions so the volume movement can be confirmed and returned to scratch in one action. NOTREADYTOSCRATCH means that the volumes are private volumes with release actions other than return to scratch. Figure 43 shows how to confirm all moves from location REMOTE to location SHELF that are ready to scratch.

```
RMM CHANGEVOLUME * CONFIRMMOVE(REMOTE,SHELF,READYTOSCRATCH)
```

Figure 43. 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.

Specify the source and target locations for the volume move by using the CHANGEVOLUME LOCATION and CHANGEVOLUME 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 44.

```
RMM SEARCHVOLUME VOLUME(*) LOCATION(LIB1) DESTINATION(LIB2) -
    INTRANSIT CLIST('RMM CHANGEVOLUME ',' CONFIRMMOVE')
```

Figure 44. Identifying Locations and Destinations for Volumes

DFSMSrmm builds a CLIST data set. Edit the CLIST data set to remove any volumes that did not get moved. Then run the CLIST 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 329 for more information on using SEARCHVOLUME with the CLIST operand. See “Creating CLISTs of Executable Subcommands” on page 128 for more information on creating lists of executable subcommands.

Chapter 5. 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 that are linked with NEXTVRS or ANDVRS. The transition through the policy takes place as DFSMSrmm processes each subchain. Inventory management and point in time status trigger the transition.

Before You Start Defining Retention and Movement Policies

This section summarizes the tasks that you should perform to implement retention and movement policies.

A Summary of the Tasks To Perform

1. Set installation-wide retention periods by using the parmlib member EDGRMMxx OPTION command.
2. Set vital record processing options by specifying the parmlib member EDGRMMxx OPTION command operands MOVEBY, RETAINBY, VRSCHANGE, VRSJOBNAME, VRSMIN, and VRSEL.
3. Define vital record specifications to retain and move data sets and volumes. DFSMSrmm can retain and move volumes in sets or as individual 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 that you review the following references before defining policies to DFSMSrmm.

- Chapter 3, “Defining Vital Record Specifications,” on page 39 and Chapter 4, “Retaining and Moving Your Volumes,” on page 55 describe the retention and movement policies you can define with DFSMSrmm. You can define the policies known as vital record specifications using the RMM ISPF dialog or the RMM TSO subcommand.

  Chapter 3, “Defining Vital Record Specifications,” on page 39 describes how to define vital record specifications by using the RMM ISPF dialog.
- Chapter 9, “Using RMM TSO Subcommands,” on page 163 describes the following RMM TSO subcommands: ADDVRS, DELETEVRS, LISTVRS, and SEARCHVRS.

  Recommendation: There is no RMM CHANGEVRS subcommand. To change a policy, first use the RMM DELETEVRS subcommand to remove the vital record specification. Then use the RMM ADDVRS subcommand 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 specifications easier.

© Copyright IBM Corp. 1992, 2004

- Chapter 7 of the *z/OS DFSMSrmm Implementation and Customization Guide* describes the DFSMSrmm parmlib member EDGRMMxx OPTION command operands. The operands CATRETPD, MAXRETPD, and RETPD are used to specify retention periods. The operands VRSCHANGE, 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. The DFSMSrmm parmlib member EDGRMMxx OPTION command RETAINBY and MOVEBY operands are used to specify if DFSMSrmm retains or moves volumes as a set or as individual volumes. See "Retaining and Moving Volumes as Sets or Individually" on page 61 for information about using the DFSMSrmm EDGRMMxx parmlib OPTION RETAINBY and MOVEBY operands to define how volumes are retained or moved.
- Chapter 13 of the *z/OS DFSMSrmm Implementation and Customization Guide* describes DFSMSrmm inventory management vital record processing that must be performed for retention and movement policies to take effect.
- Chapter 11 of the *z/OS DFSMSrmm Implementation and Customization Guide* provides vital record specification examples for retaining and moving DFSMShsm™ data sets.
- Chapter 10 of the *z/OS DFSMSrmm Implementation and Customization Guide* 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, continue to use VRSEL(OLD). Consider migrating to use VRSEL(NEW) described in "Migrating to Using VRSEL(NEW)" on page 88 at a later date to fully use DFSMSrmm retention and movement processing.
- 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 88 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). Once you are in production, consider migrating to use VRSEL(NEW) described in "Migrating to Using VRSEL(NEW)" on page 88.

- If you are early in your migration to DFSMSrmm or are a new user, use the latest conversion tools that will build vital record specifications ready for use with VRSEL(NEW).

Selecting VRSEL(OLD) Processing

When you specify VRSEL(OLD), you can perform these tasks.

- 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 that are based on the matching order described in Table 7 on page 44.
- 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 managed to a threshold, such as for an automated tape library.

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 each 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 managed to a threshold, such as for an automated tape library.
- 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 102 for information about release actions.
- You can request that volumes are returned to scratch status in one expiration processing run 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.
- You can use DFSMSrmm to release DFSMShsm tapes that are requested to be purged by DFSMShsm. By default, the expiration date protection for DFSMShsm tapes is done by DFSMShsm. DFSMShsm uses 1999/365 as the expiration date for permanent retention. To enable extra days retention for purged DFSMShsm tape volumes, you need to set up retention options in the vital record specifications that are used to retain the tape volumes.
- You can control how DFSMSrmm returns volumes to scratch using the ADDVRS RELEASE(EXPIRYDATEIGNORE) operand and the ADDVRS RELEASE(SCRATCHIMMEDIATE) operand. If you specify the SCRATCHIMMEDIATE operand and the only release action specified for volumes is SCRATCH, then you can return the volumes to scratch in a single run of inventory management expiration processing. If you specify the EXPIRYDATEIGNORE operand, DFSMSrmm ignores the volume expiration date when the volume last change date and time is earlier than the vital record processing run date and time. DFSMSrmm then sets the volume to pending release status.

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 described in “EDGRVCLN REXX Procedure to Clean Up Name Vital Record Specifications” on page 89 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 that 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 that 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, step 9, and step 10 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.

EDGRVCLN REXX Procedure to Clean Up Name Vital Record Specifications

Before you can use VRSEL(NEW), you need to clean up the retention information in your existing the vital record specifications. Use the EDGRVCLN REXX procedure to report and update existing vital record specifications that were created before VRSEL(NEW) was introduced. The procedure is intended for use only during the implementation of name vital record specification retention information and for cleanup of vital record specifications even if name vital record specification retention information is not being implemented. EDGRVCLN provides options that you can use to identify retention information that needs to be corrected and options that you can use to correct the information.

- Use the LIST(CYCLES) parameter if you want to convert CYCLES to BYDAYSCYCLE vital record specifications. You can use the output generated by LIST(CYCLES) to identify the vital record specifications and optionally change some vital record specifications. You can use FIX(BYDAYSCYCLE) to convert ALL CYCLE vital record specifications to BYDAYSCYCLE vital record specifications.
- Use the LIST(CURRENT) parameter to identify any vital record specifications that use the DFSMSrmm-restricted location name CURRENT. You can edit the commands produced by running EDGRVCLN to change the location name from CURRENT to a new location name.

- Use the LIST(DSNCHAIN) to list all vital record specification chains. The output includes all data set vital record specifications and all name vital record specifications in the chain. EDGRVCLN does not include a parameter to fix the vital record specifications chains. EDGRVCLN provides commands that you can use to add and delete vital record specifications if you want to make any changes to the vital record specification chains.

If you do not clean up vital record specification errors, DFSMSrmm issues error messages during inventory management vital record processing and sets a non-zero return code.

EDGRVCLN Parameters

The EDGRVCLN procedure accepts the following parameters:

LIST(DSNCHAIN)
Use this parameter to list all vital record specification chains.

The procedure searches for all data set vital record specifications. The procedure then searches for all name vital record specifications in the chain.

LIST(CURRENT)
Use this parameter to list all vital record specifications that specify LOCATION(CURRENT).

The procedure searches for all vital record specifications and lists each one that includes the LOCATION(CURRENT).

LIST(CYCLES)
Use this parameter to list all vital record specifications that specify a retention type of CYCLES. This includes those vital record specifications that include WHILECATALOG where CYCLES is used as the default retention type.

The procedure searches for all data set vital record specifications and lists each one that includes the CYCLE retention type.

LIST(ERROR)
Use this parameter to list all name vital record specifications that contain incorrect or incomplete retention information. The procedure assumes that any name vital record specification containing a retention type or count value is in error. This is the default value.

The procedure searches for all name vital record specifications and lists each one that includes any retention type or count value.

FIX(ERROR)
Use this parameter to correct all name vital record specifications that contain incorrect or incomplete retention information as found by LIST(ERROR). The corrections are made by deleting and re-adding the vital record specifications that contain errors. Only use this option when you are ready to correct the errors listed by the LIST(ERROR) option.

FIX(CYCLEBYDAYS)
Use this parameter to change all CYCLES vital record specifications to use the CYCLEBYDAYS retention type.

EDGRVCLN LIST File

When you use one of the LIST parameters, the procedure produces the DFSMSrmm ADDVRS subcommands that can be issued to define corrected vital record specifications. When you use the LIST(CURRENT), LIST(ERROR), or LIST(CYCLES) parameters, the LIST file includes an RMM DELETEVRS subcommand for each vital record specification in addition to the ADDVRS subcommand. If you do not want to use the FIX parameters to correct all the

identified vital record specifications, you can use the LIST file as input to your own processing. You can edit the file to remove or modify the commands to meet your specific requirements.

Running EDGRVCLN

Before making any changes using the FIX parameters, ensure that you have a valid backup of the DFSMSrmm control data set. Also ensure that you have used the LIST parameters to produce and check a LIST file because the procedure deletes and adds vital record specifications as part of the FIX processing. Use the JCL in Figure 45 to run the procedure in batch. You can also use the TSO subcommands from the SYSTSIN file interactively. Two ALLOC commands are used. One ALLOC command with MOD is treated as NEW if the data set does not exist. The second ALLOC command changes MOD to SHR if the data set existed previously which allows the data set to be reused.

```
//TMP EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSPROC  DD DISP=SHR,DSN=SYS1.SEDGEXE1
//SYSTSIN  DD *
ALLOC FILE(LIST) MOD CATALOG DA(EDGRVCLN.LIST) UNIT(SYSDA) -
 SPACE(3 1) TRACKS RECFM(V B) LRECL(255) REUS
ALLOC FILE(LIST) SHR DA(EDGRVCLN.LIST) REUS
%EDGRVCLN LIST(ERROR)
/*
```

Figure 45. Sample JCL to Run the EDGRVCLN Procedure

To clean up name vital record specifications:

1. Run EDGRVCLN with the LIST(ERROR) parameter.
2. Check the errors reported in the LIST file. If you have converted to DFSMSrmm from another product, the errors were likely introduced by the conversion programs.
3. Ensure there is a current backup of the DFSMSrmm control data set.
4. Clean up the name vital record specifications. Use the FIX(ERROR) parameter or run the DFSMSrmm subcommands in the LIST file. You can use the LIST file if you have made any modifications to the commands the procedure generated.
5. If recovery of the control data set is necessary, restore from the backup, but do not forward recover from the journal as this will repeat the vital record specification changes you want to back out. If you plan to back out changes, you should stop tape processing during the cleanup exercise.

Chaining Retention and Movement Policies

A vital record specification chain is a data set vital record specification 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 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. You can use the retention policy operands to manage all retention and to override any retention policy specified by command or JCL. For example, you

can use the RMM ADDVRS RELEASE(EXPIRYDATEIGNORE) operand to override long retention periods that your users might have set in JCL. If you want to allow users to use JCL to specify retention dates, do not use the RELEASE(EXPIRYDATEIGNORE) operand in your vital record specifications.

- 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.

You might find using the DFSMSrmm ISPF dialog easier than using the RMM TSO subcommands because the information in each panel is grouped as retention, movement, or vital record specification management fields.

For example, Figure 46 shows two vital record specifications for retaining DFSMShsm data sets.

Figure 47 and Figure 48 on page 94 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 46. 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
                                Release options:
Next VRS in chain . . HSMVRS01  Expiry date ignore  . . . . yes
    Chain using . . . NEXT      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 47. Specifying a Vital Record Specification Using the Dialog Part 1

Figure 48 on page 94 shows the same information as shown in the second vital record specification defined in Figure 46.

```
  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 48. 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 DFSMSrmm 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, DFSMSrmm issues message EDG2229I, and DFSMSrmm 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.

We recommend that you define a global policy that retains data sets for a specified period, such as only as long as they are cataloged. The RMM ADDVRS subcommand you issue is:

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

With VRSEL(NEW), DFSMSrmm can match two vital record specifications to a data set. 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 default retention values.

NOGDG is the default. If you specify NOGDG on '**' or '*.**' vital record specifications, DFSMSrmm retains all data sets including generation data sets that do not match any other vital record specification. If you want to define a vital record specification that retains all GDGs, you can use the DFSMSrmm pseudo-GDG mask described in "Pseudo-GDG Data Set Names" on page 48.

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 catalogued, 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 the data sets 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                    */
           EXTRADAYS           -          /* 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 DSN('A.**')      -             /* Data set VRS                  */
RMM ADDVRS DSN('A.**')      -             /* Data set VRS                  */

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 the data set is retained for 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 set 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 6. 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 281 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 that 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.
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.

© Copyright IBM Corp. 1992, 2004

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 186 and “CHANGEVOLUME: Changing Volume Information” on page 234 for information on using the ADDVOLUME and CHANGEVOLUME subcommands. Use the ADDVRS subcommand to specify release actions in vital record specifications. See “ADDVRS: Adding a Vital Record Specification” on page 205 for information about the ADDVRS subcommand.

A master volume 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 or you have requested that DFSMSrmm should ignore the expiration date. You can request that DFSMSrmm should ignore the expiration date by specifying the RMM ADDVRS RELEASE(EXPIRYDATEIGNORE) operand.
- 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.

Related Reading:

- See “ADDVOLUME: Adding Volume Information” on page 186 and “ADDVRS: Adding a Vital Record Specification” on page 205 for information about defining policies to control when volumes are eligible for release.
- See “DELETEVOLUME: Deleting Volume Information” on page 275 for information about using the DELETEVOLUME subcommand to manually release one or more volumes.
- See “Deleting Information for Volumes in Multivolume Sets” on page 31 for information about using the DFSMSrmm ISPF dialog to release multiple volumes in a multi-volume set.

Setting Release Actions for a Volume

To set release actions for a volume:

- Use the Add Volume panel or the RMM 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 to set release actions, 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. Table 13 describes the release actions you can specify for a volume:

Table 13. Release Actions for a Volume

Release Action	Type of Processing
Any one of the following:	
• Returning volumes to scratch status	Automatic when EDGHSKP 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

Returning Volumes to Scratch Status—Automatic Action

You can specify that DFSMSrmm is to automatically returns the volume to scratch status, once any other release actions specified for the volume are completed. Returning a volume scratch status is mutually exclusive with replacing a volume and returning a volume to an 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

You can set the release action for a volume so that the volume is replaced when the volume becomes eligible for release.

You replace volumes when the volumes have permanent I/O errors or when the volumes 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 329 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

You can set up DFSMSrmm to mark the volume 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 that 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 107 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 329 for more information on the SEARCHVOLUME subcommand.

Initializing and Erasing Volumes—Automatic Action

You can set up DFSMSrmm to automatically erase and initialize a volume when the volume is released. You can also specify either release action separately. Because these actions require that volumes be mounted, specify initialize or erase 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 *z/OS 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.

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.

Restriction: Do not degauss 3590 media as the servo tracks will be destroyed.

To create a list of volumes that need to be erased or initialized, 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 329 for more information on the SEARCHVOLUME subcommand.

Notifying Owners—Automatic Action

You can set up DFSMSrmm to send a message to the volume’s owner when DFSMSrmm marks the volume as pending release. In addition to setting the release action, you must set the DFSMSrmm parmlib OPTION NOTIFY operand to Y, and define a valid user ID and node for the owner.

For information on the message content and how to modify it, and for information on setting parmlib options, see *z/OS 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 281, “ADDVOLUME: Adding Volume Information” on page 186, and “CHANGEVOLUME: Changing Volume Information” on page 234 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 32, “ADDOWNER: Adding Owner Information” on page 176, or “CHANGEOWNER: Changing Owner Information” on page 229 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 329 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 that is 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 109 for details on manually releasing volumes. You can also use the DELETEVOLUME subcommand with the RELEASE operand; see “DELETEVOLUME: Deleting Volume Information” on page 275 for more information.

Defining Vital Record Specifications Release Options

You can define data set vital record specifications with the RELEASE operand to control the processing DFSMSrmm performs when a volume is no longer retained by a vital record specification. Use the RELEASE(EXPIRYDATEIGNORE) operand to release a volume even the volume expiration date has not been reached. Use the RELEASE(SCRATCHIMMEDIATE) operand if you want to return a volume to

scratch status immediately in a single run of DFSMSrmm inventory management. To use these values, you must specify the DFSMSrmm EDGRMMxx parmlib OPTION command VRSEL(NEW) operand. Use the ADDVRS subcommand or the DFSMSrmm ISPF dialog to define vital record specification. See “ADDVRS: Adding a Vital Record Specification” on page 205 or Chapter 4, “Retaining and Moving Your Volumes,” on page 55 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 205 or Chapter 4, “Retaining and Moving Your Volumes,” on page 55 for more information.

Changing the Expiration Date for a Volume

DFSMSrmm records the expiration date for a volume 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 that are 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 234 or “Changing Volume Information” on page 29 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:

- To return volumes to scratch status
- To erase and initialize volumes
- Notification of 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.

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.
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
 -- -------- -------- -------- --------- -------
 S  REPLACE                               Pending
```

2. Press ENTER.

   DFSMSrmm displays the DFSMSrmm Volume Action Summary which contains a list of volumes.

3. Table 14 describes the line operators you can use against specific entries in the list to confirm pending release actions:

Table 14. Confirm List Line Operators

Line Operator	Description
C	Confirm any release action
CA	Confirm any volume movement 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
CS	Confirm that the volume has been returned to scratch
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 49 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 49. Confirming Replace Actions for Volumes

See “SEARCHVOLUME: Creating a List of Volumes” on page 329 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 275 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 102 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. They should use the Release Volumes panel available to them from the User Menu. See “Releasing Volumes Manually” on page 19 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.
2. Specify a volume serial number or leave blank and DFSMSrmm creates 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. The panel displays information about the single volume you are releasing and asks you to confirm that you want to release the volume.
   - 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, 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.
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 29 for more information.
5. If you requested that DFSMSrmm build a list of volumes, DFSMSrmm displays the DFSMSrmm Volumes panel which contains a list of volumes.

   Use the line operators to perform actions on the volumes in the list. Table 15 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 15.

Table 15. 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 7. 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 dropship 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. Volumes must be returned from storage locations and loan locations 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 *z/OS DFSMSrmm Implementation and Customization Guide* for information on scheduling inventory management utilities. See *z/OS DFSMSrmm Reporting* for more details on EDGRPTD and reports that 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.

© Copyright IBM Corp. 1992, 2004

Table 16 describes what to specify on the different display panels to request information about the resources defined to DFSMSrmm.

Table 16. 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.
Volume	A specific volume serial number You can also limit the amount of information DFSMSrmm displays about the volume by specifying which of four panels you want to view.
Software product	A specific software product number You can also specify a version of the software product.
Data set	A specific data set name and volume serial number 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.
Vital Record Specification	Either a data set name or a volume serial number You can use a data set name mask or a generic volume serial number to request information about a vital record specification.

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 50 on page 113.

```
  Panel  Help
------------------------------------------------------------------------------
EDGPT100                  DFSMSrmm Volume Display
Command ===>

Volume serial . . . A00001









Enter the volume serial and press ENTER to display volume information.
```

Figure 50. 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 50.
3. Press ENTER.

   DFSMSrmm displays a panel containing information about the volume you specified.

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 *z/OS DFSMSrmm Implementation and Customization Guide* for information.

Related TSO Subcommand

Use the LISTCONTROL subcommand with the OPTION operand to request information about system options that are defined by your installation. See "LISTCONTROL: Displaying Parmlib Options and Control Information" on page 287 for more information.

To display the panel shown in Figure 51 on page 114, you could enter a fast path command from the command or option line of any panel.

```
  Panel  Help
------------------------------------------------------------------------------
EDGPC200              DFSMSrmm System Options Display
Command ===>


 Parmlib suffix . : WT

 Operating mode . : PROTECT

 Data sets:
  Control  . . . : RMMTST.LARGE.MASTER
  Journal  . . . : RMMUSER.APAR.JOURNAL
  CDS id . . . . : SJTEST                  Journal threshold . : 50 %
  Catalog SYSID  : NOTSET

Retention period:                          SMF:
  Default  . . . : 5                         System id . . . . : W98MVS1
  Maximum  . . . : 9999                      Audit . . . . . . : 248
  Catalog  . . . : 12     hours              Security  . . . . : 249

PDA: ON                                    Report options:
  Block count  . : 255                       Lines per page  . : 54
  Block size . . : 27                        Date format . . . : AMERICAN
  Log  . . . . . : ON

Miscellaneous:                             Auto-start procedures:
  RACF support . . . . . . : AUTOMATIC       Scratch procedure : EDGXPROC
  MAXHOLD limit  . . . . . : 250             Backup procedure  : SUB
  IPL check  . . . . . . . : NO
  Uncatalog  . . . . . . . : SCRATCH       VRS:
  System notify  . . . . . : YES             Job name  . . . . : 2
  BLP  . . . . . . . . . . : RMM             Minimum count . . : 1
  Master overwrite . . . . : MATCH           Minimum action  . : FAIL
  Message case . . . . . . : MIXED           Change  . . . . . : INFO
  Accounting . . . . . . . : JOB             VRSEL processing  : OLD
  Disposition DD name  . . :                 Retain by . . . . : VOLUME
  Disposition message ID . : EDG4054I        Move by . . . . . : VOLUME
  PREACS . . . . . . . . . : NO
  SMSACS . . . . . . . . . : NO            SMSTAPE:
  TVEXT purge  . . . . . . : RELEASE         Purge . . . . . . : ASIS
  Reuse bin  . . . . . . . : CONFIRMMOVE     Update exits. . . : YES
  COMMANDAUTH owner  . . . : NO              Update commands . : YES
  COMMANDAUTH dsn  . . . . : NO             Update scratch. . : YES
  Default media name . . . : 3480
  Local tasks  . . . . . . : 10

Client/Server:
  System type  . . . . . . : STANDARD
  Server tasks . . . . . . : 0
  Server Port number . . . : 0
  Host name  . . . . . . . :
  IP address . . . . . . . :
Enter SCROLL commands for more information, or END to CANCEL
```

Figure 51. DFSMSrmm System Options Display

Using Resource Lists

Any user can request a list of resources that are 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 117 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 123 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 9, “Using RMM TSO Subcommands,” on page 163 for more information on the SEARCH subcommands.

DFSMSrmm allows you to specify generic names or numbers as search criteria. Therefore 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 17 describes how to select the various DFSMSrmm ISPF dialog search panels.

Table 17. How to Display Search Panels

To Display a Search Panel for:	Do One of the Following:	And DFSMSrmm Displays:
Shelf locations	• 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.	The DFSMSrmm Rack or Bin search panel.
Volumes	• 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.	The DFSMSrmm Volume search panel.
Software products	• 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.	The DFSMSrmm Product search panel.
Data sets	• 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.	The DFSMSrmm Data Set search panel.

Table 17. How to Display Search Panels (continued)

To Display a Search Panel for:	Do One of the Following:	And DFSMSrmm Displays:
Vital record specifications	• 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.	The DFSMSrmm VRSs search panel.

2. Press ENTER.
3. Specify the search criteria you want DFSMSrmm to use to create the list. Press PF1 for field-specific help.

   For example, to create a list of all empty shelf locations in the removable media library, use the Rack and Bin Search panel. 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 that are 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 that are 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 117 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 123 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 by 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*).

- Use one of the RMM TSO SEARCH subcommands instead of a panel to request the search. The results are in linemode messages. Use the SAVE ON command to process the subcommand in batch mode. See Chapter 1, “Getting Started with DFSMSrmm,” on page 1 for information on the SAVE ON command.

How to Sort a List

You can change the sort order DFSMSrmm uses to build lists by:

- The use of 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.
- Type **OPTIONS SORT** on the command line of any panel and press ENTER.

DFSMSrmm displays the Dialog Sort Options Menu as shown in Figure 52:

```
  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.









5694-A01 (C) COPYRIGHT 1993,2004 IBM CORPORATION
```

Figure 52. 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 53 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 53. 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 the descending order. DFSMSrmm uses the initial setting or the most recent value that is 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 18 describes the different table field names you can sort on and identifies the lists in which they are used.

Table 18. 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

Table 18. Table Field Names on DFSMSrmm Lists (continued)

Table Field Name	Sort Command Parameter	Description	Type of List
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, *library_name*)	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, *library_name*)	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
Name	Name	Report name	Report
Next VRS Name	Nextvrs	Name of next vital record specification to which this vital record specification is linked	Vital record specification
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

Table 18. Table Field Names on DFSMSrmm Lists (continued)

Table Field Name	Sort Command Parameter	Description	Type of List
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
Title	Title	Report title	Report
Type	Type	Type of vital record specification (data set, volume, or name) and reporting type	Vital record specification and reporting type
Untilexpired	Untilexpired	Retention type	Vital record specification
Userid	Userid	Report user ID	Report
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 the 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 54 describes the syntax to sort a list of rack or bin numbers.

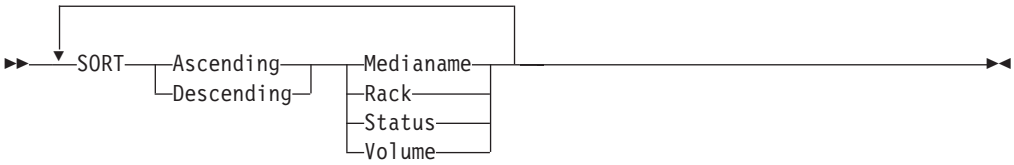


Figure 54. SORT Command Syntax Diagram for Rack or Bin Number Lists

Volume Lists

Figure 55 on page 122 describes the syntax to sort a list of volumes.

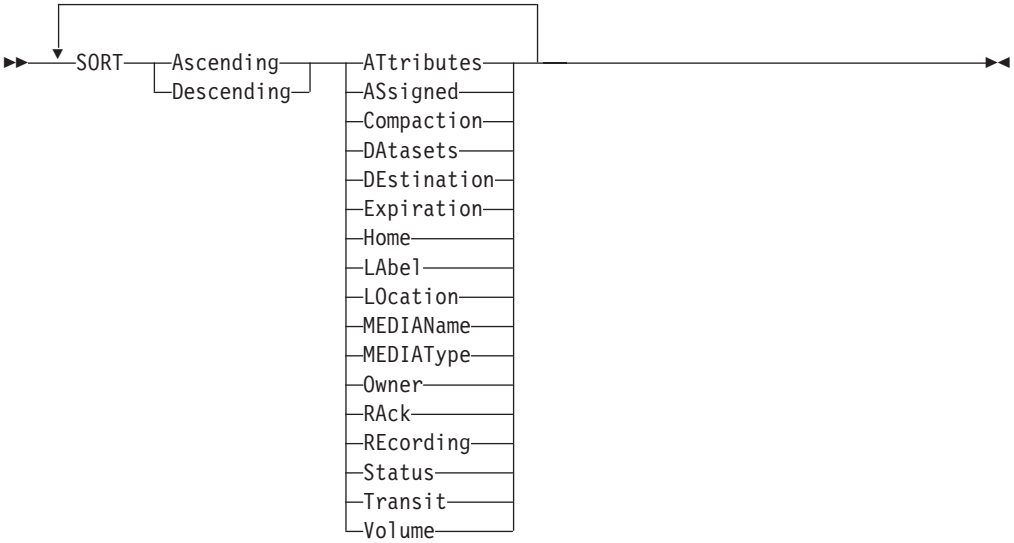


Figure 55. SORT Command Syntax Diagram for Volume Lists

Volume Movements and Actions Lists

Figure 56 describes the syntax to sort a list of volume movements and actions.

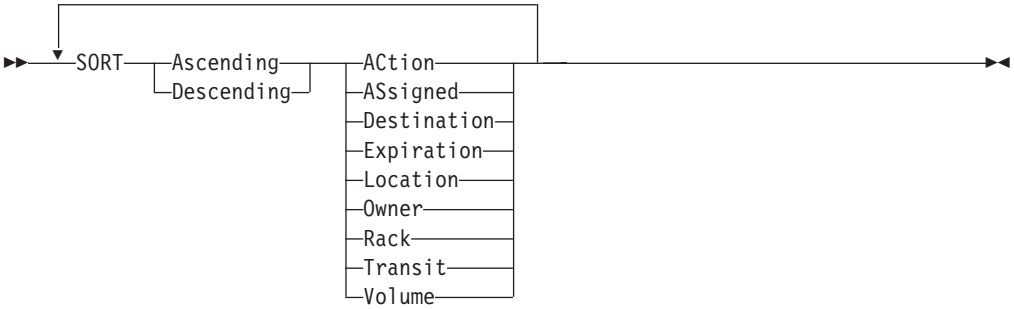


Figure 56. SORT Command Syntax Diagram for Lists of Volume Movements and Actions

Software Product Lists

Figure 57 describes the syntax to sort a list of software products.

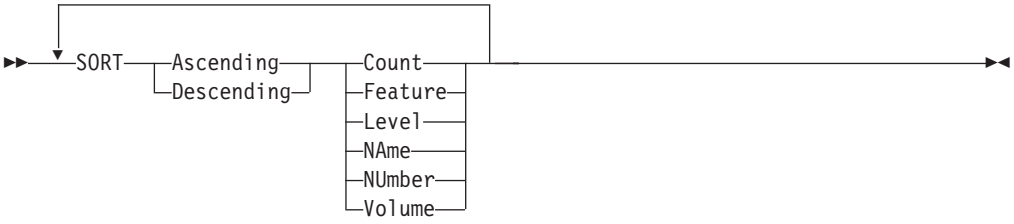


Figure 57. SORT Command Syntax Diagram for Software Product

Data Set Lists

Figure 58 describes the syntax to sort a list of data sets.

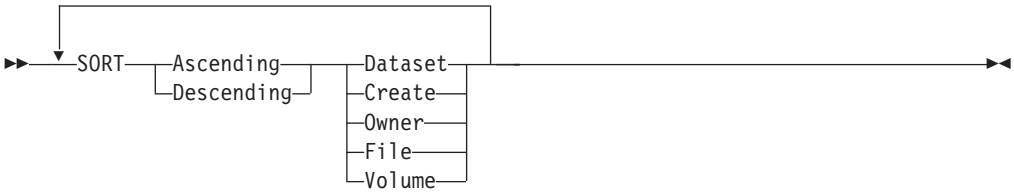


Figure 58. SORT Command Syntax Diagram for Data Set Lists

Vital Record Specification Lists

Figure 59 describes the syntax to sort a list of vital record specifications. Once you change the initial settings for sort values, DFSMSrmm stores the most recently specified values and remembers them across sessions.

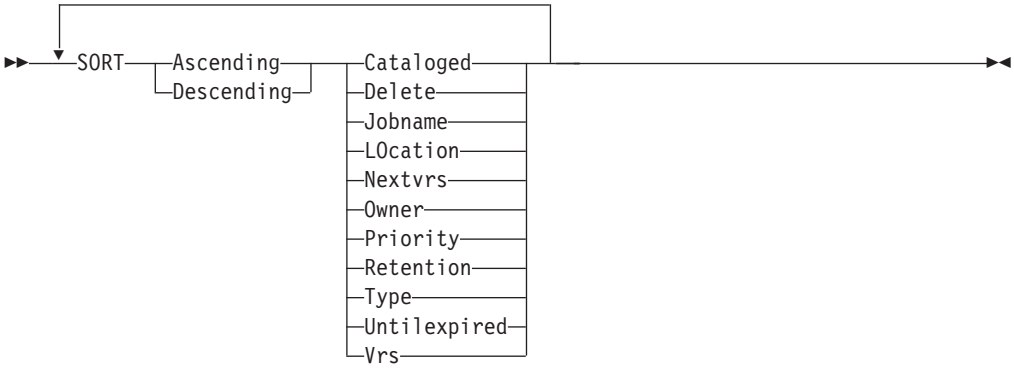


Figure 59. SORT Command Syntax Diagram for Rack or Bin Number Lists

Report Generator Lists

Figure 60 describes the syntax to sort a list of reports.

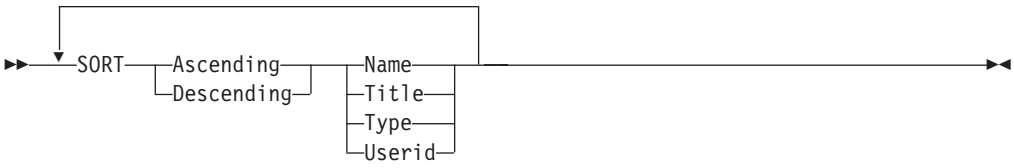


Figure 60. SORT Command Syntax Diagram for Report Generator Lists

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 enter line operators on the DFSMSrmm Rack or Bin panel to obtain information about shelf locations. See the DFSMSrmm ISPF dialog help for details about the line operators that you can use for shelf location lists.

Volume Lists

You can use enter line operators on the DFSMSrmm Volumes panel to obtain information about volumes or to make changes for volumes in the list if you have appropriate authority to make changes. See the DFSMSrmm ISPF dialog help for details about the line operators that you can use for volume lists.

You can obtain volume set information from the results of a volume search. You can also create a list of volumes ready for release from the results of a volume search.

Software Product Lists

You can use enter line operators on the DFSMSrmm Products panel to obtain information about software products and associated volumes or to make changes for the software products and volumes in the list if you have appropriate authority to make changes. See the DFSMSrmm ISPF dialog help for details about the line operators that you can use for product lists.

Data Set Lists

You can use enter line operators on the DFSMSrmm Data Sets panel to obtain information about data sets or to make changes for the data sets in the list if you have appropriate authority to make changes. See the DFSMSrmm ISPF dialog help for details about the line operators that you can use for data sets.

Vital Record Specification Lists

You can use enter line operators on the DFSMSrmm VRSs panel to obtain information about vital record specifications or to make changes for the vital record specifications in the list if you have appropriate authority to make changes. See the DFSMSrmm ISPF dialog help for details about the line operators that you can use for vital record specifications.

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 SEARCHVOLUME subcommand or the Volume Search panel to request a list of available scratch volumes to be pulled from their shelf locations and mounted for use.

Using the RMM SEARCHVOLUME Subcommand

To create a volume pull list, you can issue the RMM SEARCHVOLUME subcommand as follows:

```
==> RMM SEARCHVOLUME VOLUME(*) STATUS(SCRATCH) TYPE(PHYSICAL) OWNER(*) LIMIT(*)
```

See “SEARCHVOLUME: Creating a List of Volumes” on page 329 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 61 produces a list of up to 100 scratch volumes. 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 *
 SEARCHVOLUME VOLUME(*) STATUS(SCRATCH) TYPE(PHYSICAL) OWNER(*) LIMIT(100)
 /*
```

Figure 61. Producing a List of Scratch Volumes

Using the Volume Search panel

You can also use the Volume Search panel, shown in Figure 62.

```
  Panel  Help  Scroll
--------------------------------------------------------------------------------
EDGPT010                  DFSMSrmm Volume Search
Command ===>

                                                                    More:     +
Volume  . . . .  *        May be generic.  Leave blank for all volumes.

Owner . . . . .  *        Owned by a specific user. Default is your userid

Limit . . . . .  *        Limit search to first nnnn volumes

Media name  . .           Limit to a single media name
Pool prefix . .           or to a particular pool

Status  . . . . A
                          Select volume status.  Default is ALL.
Since . . . . .           Volumes assigned since YYYY/DDD
Retention . . .           Volumes retained up to YYYY/DDD
Clist . . . . .           YES to create a data set, or NO, or blank

Home  . . . . .           Limit to volumes with this home location name
Location  . . .           Limit to volumes in a single location
In container              Stacked volser

Volume type . .           ( LOGICAL , PHYSICAL or STACKED )
Media type  . .           Tape media type ( for example, HPCT )
Label . . . . .           Tape label standard ( for example, SL )
  Current version         Label version number( for example, 3 )
  Required version        Label version number( for example, 4 )
  VOL1 volser . .         Volser in the VOL1 label ( NONE or volser )
Density . . . .           Tape recording density
Format  . . . .           Tape recording format ( for example, 256TRACK )

Compaction  . .           Limit to volumes containing compacted data
Attributes  . .           Tape special attributes ( NONE or RDCOMPAT )

Destination . .           Limit by destination
Required  . . .           Limit by required location
Move mode . . .           Limit by move mode ( AUTO or MANUAL )
Intransit . . .           Limit to volumes which are moving ( YES or NO )

Enter HELP or PF1 for the list of available line commands
Enter SCROLL commands for more search values, ENTER to SEARCH or END to CANCEL
```

Figure 62. DFSMSrmm Volume Search Panel

DFSMSrmm displays a list of scratch volumes. DFSMSrmm lists summary information for each item in the list, including an optional rack number. Use this list to locate and pull the scratch volumes you need. Place the scratch volumes near the drives where you use them or into cartridge loaders.

Dropship Lists

When you move volumes between locations in the removable media library or between storage locations, use one of the following to request a dropship 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 329 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 63 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 63. DFSMSrmm Volume Action Status Panel

2. Press ENTER.

   DFSMSrmm returns a list in panel EDGPT610 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 79 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 287 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 DFSMSrmm Control Information Menu panel.

- Select Option 4 (VLPOOLS) on the Control Information Menu and press ENTER.

  DFSMSrmm displays the DFSMSrmm Volume Pool Definitions panel.

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 DFSMSrmm Volume Action Status panel.

1. Type `S` against the type of volume move or release action for which you want a list of volumes.
2. Press ENTER.

   DFSMSrmm displays the DFSMSrmm Volume Action Summary List which contains a list of volumes.
3. Use the ISPF command, PRINT or PRINT-HI, to print the list.

   See Table 14 on page 108 for the list of line operators you can use against entries in the list to confirm one or more of the outstanding moves or actions. See “Confirming Volume Movements to DFSMSrmm” on page 79 for more information.

Creating CLISTs of Executable Subcommands

You can use the DFSMSrmm ISPF dialog or any of the RMM TSO SEARCH subcommands with the CLIST operand to create a CLIST data set of executable subcommands for resources. You can modify the CLIST and save it for later use. For example, you can create a list of subcommands to confirm volume moves or release actions that have been completed. You can create a list of volumes moving out of a specific automated tape library that must be ejected. You can use the CLIST when you are ready to confirm the movement of the volumes.

You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). These text strings cannot exceed 255 characters. Separate the *prefix_string* and *suffix_string* using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns.

The data set used for the CLIST output must have variable length records and a maximum logical record size of 255. The record length includes a 4-byte length field at the front of the record. DFSMSrmm ignores the DCB attributes of the CLIST data set. If you do not specify variable length and an LRECL of 255, DFSMSrmm changes the CLIST data set attributes to variable blocked with a maximum logical record size of 255. If the length of the output record exceeds 255 characters, DFSMSrmm splits the record into multiple records and adds a continuation character, +, to all but the last record.

You can use any data set to receive the CLIST output by using the RMMCLIST DD name. If you do not allocate the RMMCLIST DD, DFSMSrmm allocates a data set called '*prefix*.EXEC.RMM.CLIST'. The value for *prefix* can be:

- A TSO user PROFILE PREFIX value which is your TSO user ID if you do not change the PROFILE PREFIX value.
- The RACF user ID from the ACEE if you have specified PROFILE NOPREFIX and the RACF user ID is valid.
- The job name when no TSO prefix value or RACF user ID is available.

When DFSMSrmm allocates this data set and determines that it is a partitioned data set, DFSMSrmm writes the output to the member TEMPNAME.

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.

Follow these steps when you use the DFSMSrmm ISPF dialog to create the CLIST data set.

1. Enter the RMM TSO command prefix and suffix information, CLIST data set name, and CLIST data set size on the DFSMSrmm CLIST processing panel.
2. Press enter. DFSMSrmm displays the DFSMSrmm CLIST Menu panel.
3. Enter E on the DFSMSrmm CLIST Menu panel to edit the saved CLIST data set or S to submit the job. You can optionally tailor the JCL prior to submitting the job on the DFSMSrmm CLIST Menu panel.

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 *z/OS ISPF User's Guide Volume I, SC34-4822* for more information.

Chapter 8. DFSMSrmm Operator Procedures

This topic 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

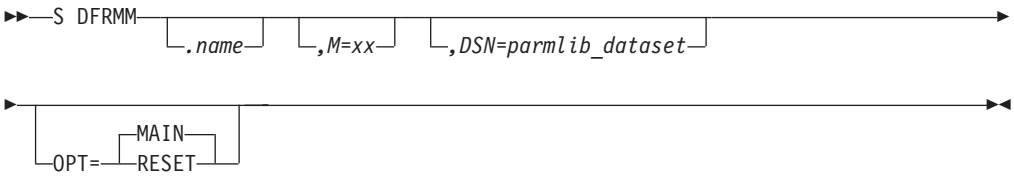
Restarting the DFSMSrmm Subsystem

Under normal circumstances, the DFSMSrmm subsystem starts automatically through standard initial program load (IPL) procedures or IPL procedures your installation has modified. In exceptional cases, such as after recovery of the DFSMSrmm control data set, you might need to restart the subsystem.

Using the START Command

Use the START command to restart the DFSMSrmm subsystem shown in Figure 64.

Figure 64. START Command Syntax Diagram



where:

DFRMM
DFRMM is the default procedure name. Your installation might have defined a name other than DFRMM. See *z/OS 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 *z/OS 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.

© Copyright IBM Corp. 1992, 2004

RESET
Use OPT=RESET to reset the DFSMSrmm subsystem interface to remove DFSMSrmm from the system on which this command is issued.

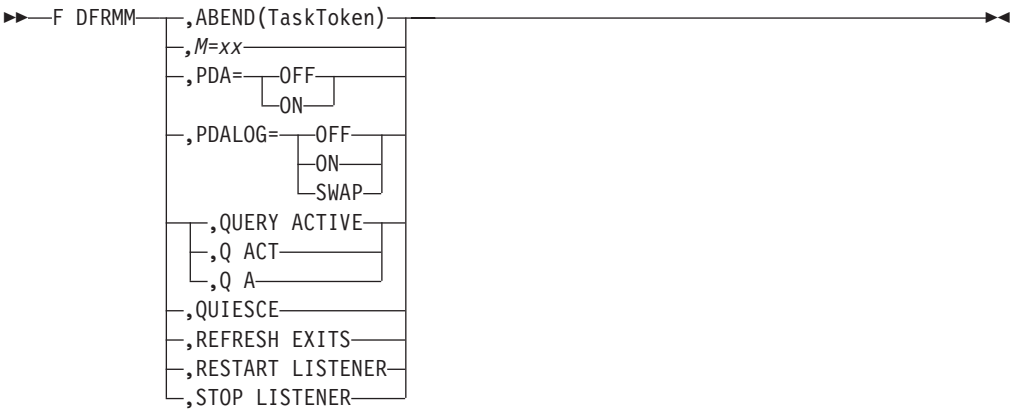
At the console, enter S DFRMM to start DFSMSrmm, using the default procedure name and parmlib member.

When the DFSMSrmm subsystem interface is enabled, tape volume usage is rejected until the DFSMSrmm subsystem is active.

Using the Modify Command

Use the MODIFY command to restart the DFSMSrmm subsystem as shown in Figure 65.

Figure 65. MODIFY Command Syntax Diagram



where:

DFRMM
DFRMM is the default procedure name. Your installation might have defined a name other than DFRMM. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information on defining the started procedure name.

,ABEND(TaskToken)
Specifies that you want to cancel a subsystem request processing task or an IP request subtask. The *TaskToken* is a token from the system on which the task is running that identifies the subtask handling the request. You can obtain the *TaskToken* by issuing the command

```
F DFRMM,QUERY ACTIVE
```

,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.

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 following command.

```
F DFRMM,M=xx
```

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 the *z/OS DFSMSrmm Implementation and Customization Guide* for information about defining the procedure name and about 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.

,PDA=ON|OFF

Specifies Problem Determination Aid (PDA) tracing on or off.

,PDALOG=ON|OFF|SWAP

Specifies the LOGGING function during PDA tracing.

,QUERY ACTIVE

Specifies that you want the number of requests that are waiting to be processed, the number of active requests, to determine whether DFSMSrmm is active or quiesced, and whether the journal is enabled. For each active task, DFSMSrmm issues several messages that display

- Function
- System ID
- Task name
- Time started
- Task token
- TCP/IP processing task status

Example: Issue the following command to obtain information about DFSMSrmm.

```
F DFRMM,QUERY ACTIVE



EDG1119I DFSMSrmm STATUS IS ACTIVE. JOURNAL ENABLED. LISTENER ACTIVE.
EDG1120I Function  System  Task   Name  Started   Token   Status
EDG1113I      130     SYSA JOB=WOODMWIM 02:16:40 08000001    SEND
EDG1113I      114          JOB=WOODMWSV 02:16:39 0A0001A2
EDG1114I LOCAL TASKS  10, ACTIVE   1, SERVER TASKS  10, ACTIVE   1
EDG1118I      0 QUEUED REQUESTS, INCLUDING      0 NOWAIT      0 CATALOG
EDG1121I DEBUG IS DISABLED. PDA TRACE LEVEL: 1-2-3-4
EDG1101I DFSMSrmm MODIFY ACCEPTED
```

You can also use the abbreviations `Q ACT` and `Q A`.

,QUIESCE

Issue this command to manually quiesce DFSMSrmm.

,REFRESH EXITS

Specifies that the current installation exits are to be refreshed. You will probably use this command at the request of the systems programmer to overcome a problem with the current installation exits.

Example: To refresh the DFSMSrmm installation exits and enable any exits that are currently disabled, issue the following command.

```
F DFRMM,REFRESH EXITS
```

,RESTART LISTENER
On a DFSMSrmm server, issue this command to stop the TCP/IP processing task and then to restart TCP/IP processing.

,STOP LISTENER
Issue this command to stop the DFSMSrmm TCP/IP processing task.

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. You can use the EDGRESET utility by issuing the S DFRMM,OPT=RESET command to remove DFSMSrmm functionality from the system. That means that DFSMSrmm no longer processes any subsystem requests such as TSO subcommands, and recording and validation of tape processing, unless DFSMSrmm is restarted. When you issue the RESET option in an RMMplex or in a client/server environment, the DFSMSrmm subsystem interface is removed only on the system where the RESET is issued.

If you use the DFSMSrmm EDGRESET utility to remove DFSMSrmm from the system and allow tape mounts to be processed without a media management system, 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 System Management Facility (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.

Example: To shut down the DFSMSrmm subsystem, use the MVS STOP command:

```
P DFRMM
```

Before stopping, DFSMSrmm processes all existing requests to completion. New requests are rejected. If an existing, queued request is for inventory management, it fails.

Example: To manually quiesce the DFSMSrmm subsystem, use the MVS MODIFY command:

```
F  DFRMM,QUIESCE
```

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.

In a multi-host environment, conditions, which result in an automatic quiesce of DFSMSrmm (such as control data set errors from which DFSMSrmm cannot automatically recover). cause the quiesce on all hosts sharing the control data set. Only after all hosts have successfully quiesced can the control data set be recovered. Manually issuing a DFSMSrmm quiesce affects only the host on which you issue the command. If you want all hosts quiesced, you must issue the command on each host that is sharing the control data set.

Example: To remove DFSMSrmm, use the MVS START command:

```
S DFRMM,OPT=RESET
```

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.

The RESET option affects the system on which the command is issued.

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.

Example: Use the MVS MODIFY command keyword to enable PDA tracing:

```
F DFRMM,PDA=ON
```

You can also enable or disable PDA processing by using the parmlib OPTION command PDA operands described in the *z/OS DFSMSrmm Implementation and Customization Guide*.

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 *z/OS DFSMSrmm Diagnosis Guide* for details.

Responding to DFSMSrmm Operator Messages

DFSMSrmm issues several messages requiring operator intervention. If your installation has developed procedures for responding to these messages, you should follow those procedures.

The messages listed in Table 19 are identified by type and number.

Table 19. Operator Messages

Type of Message	Message Numbers
Initialization messages. See “Initialization Messages” on page 137.	EDG0001I, EDG0204I, EDG0110D, EDG0114I, EDG0105I, EDG0101I, EDG0104E, EDG1001A, EDG0103D, EDG0123D, EDG0215D, EDG0228E, EDG1101I, EDG0115D, EDG0353I, EDG0358D, EDG0361D, EDG1105I, EDG1106I, EDG1107D, EDG9115I
Tape mount messages. See “Tape Mount Messages” on page 140.	EDG4013I, EDG6627A, EDG6628A, EDG6663D, IEF233A, EDGV01D
Tape validation messages. See “Tape Validation Messages” on page 142.	EDG4021I, EDG4023I, EDG4024I, EDG4025I, EDG4026I, EDG4027I, EDG4028I, EDG4032I, EDG4033I, EDG4035I, EDG4036I, EDG4041I
Tape processing messages. See “Tape Processing Messages” on page 145.	EDG4000D, EDG4001D , EDG4005E, EDG4006E, EDG4007E, EDG4008A, EDG4010D, EDG4012D

Table 19. Operator Messages (continued)

Type of Message	Message Numbers
System error messages. See "System Error Messages" on page 146.	EDG0154I, EDG2103D, EDG2104E, EDG2107E, EDG2108E, EDG2110I, EDG2111I, EDG2112I, EDG2113I, EDG2114I, EDG2115I, EDG2116A, EDG9116I
Installation exit messages. See "DFSMSrmm Installation Exit Messages" on page 148.	EDG0300I, EDG0301I, EDG0302I, EDG0303D, EDG0304I, EDG0305I, EDG8121D, EDG8122D, EDG8123D
Tape Labeling Messages. See "Replying to LABEL Procedure Messages" on page 155.	EDG6619I, EDG6620I, EDG6621E, EDG6622I, EDG6623I, EDG6625I, EDG6626A, EDG6627A, EDG6628A, EDG6631I, EDG6642I, EDG6658I, EDG6661E,EDG6662E, EDG6663E, EDG6672E

For a complete description of the DFSMSrmm messages, see "Using LookAt to look up message explanations" on page xviii.

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 134 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.

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.

EDG0123D *function* **FOUND TO BE ACTIVE ON SYSTEM** *system_name* **- REPLY ″Y″ TO RESET STATUS OR ″N″**

– DFSMSrmm issues this message if inventory management or backup is running 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.

EDG0228E REUSEBIN(STARTMOVE) REQUIRES EXTENDED BIN ENABLED - USING (CONFIRMMOVE) - REPLY ″CONTINUE″ OR ″CANCEL″

– DFSMSrmm is processing the OPTION command in parmlib, and checking if the extended bin support is enabled.
– Report the error to the system programmer. Reply *CONTINUE* if you want to start DFSMSrmm with parmlib option REUSEBIN(CONFIRMMOVE). Reply *CANCEL* to stop DFSMSrmm.

EDG1101I DFSMSrmm *command* **COMMAND ACCEPTED**

– A stop or modify command was issued and completed. You do not need to reply to this message.

EDG0353I SERVER LISTENER TASK HAS ENDED - USE THE MODIFY COMMAND WITH ″RESTART LISTENER″ TO RESTART

– The DFSMSrmm server subsystem TCP/IP listener task has ended. If you require the DFSMSrmm subsystem to run as a server use the `F DFRMM,RESTART LISTENER` operator command once the errors are corrected.

EDG0358D SERVER servername COMMUNICATION ERROR - REPLY ″CANCEL″, OR ″RETRY″ OR ″M=xx″

– The DFSMSrmm subsystem is communicating between the DFSMSrmm client and server. Reply to EDG0358D with one of the valid responses:

Reply	And the result is
CANCEL	The current request fails. If the current request is to connect to the server at DFSMSrmm subsystem startup, replying CANCEL stops the DFSMSrmm subsystem.
RETRY	Before you RETRY, correct the error that prevented the client system from connecting to the server. DFSMSrmm retries connecting to the server and attempts to continue processing the current request.

EDG0361D SERVER STARTUP ERROR - REPLY ″RETRY″ OR ″IGNORE″

– The DFSMSrmm server subtask is starting up and has been unable to set up TCP/IP communication. Reply RETRY when you want DFSMSrmm to try starting up the server system again by retrying TCP/IP communications. Reply IGNORE when it is acceptable for the DFSMSrmm subsystem to run as a standard system and not as a server system.

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.

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 *z/OS 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 the rack number for a volume 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:

EDG4013I M *dev*,*volser*,*jobname*,*procname*, *stepname*,*rack*,*location*,*loc_bin*,*destination*,*dest_bin*,*loan_location*

– DFSMSrmm intercepted a mount request for a specific volume. The volume is probably not currently in the library. Use the information in the message to help you locate the volume. You do not need to reply to this message.

  Any null field in the message indicates that DFSMSrmm has no information for that value.

– If the volume cannot be mounted, cancel the job.

EDG6627A M *drive_number* **VOLUME(***volser***) RACK(***rack_number***) TO BE** *erased_or_labelled_or_verified vol1_volser, 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 *dev,ser,[labtyp],jobname [,stepname] [,dsname],* **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:

  dev is the device number.
  ser is the volume serial number.
  labtyp is the type of label: SL for standard label, or NL for no label or by-pass label.
  jobname is the name of the job.

stepname is the name of the step.
dsname is the data set name.
rackno is the external tape label.
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.

For example:

```
IEF233A M BE1,PRIVAT,SL,PRITCDSX,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,PRITCDSX,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 *z/OS DFSMSrmm Implementation and Customization Guide* for information about DFSMSrmm tape mount validation rules.

DFSMSrmm issues a tape rejection message 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 that indicates the reason for the rejection, and the user can refer to the

message explanation for help in solving the problem. When you are running DFSMSrmm in warning mode, DFSMSrmm can issue up to two sets of messages when a volume is being rejected.

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 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 *z/OS 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. FIRST FILE HDR1 NAME** *hdr1_info* **DOES NOT MATCH RECORDED NAME** *first_dsname*

– 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, EDG4000D, EDG4001D and EDG4010D, describe errors that can occur before tape validation occurs.

EDG4000D JOURNAL FILE IS LOCKED 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 backed up and the journal file cleared.

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 do not 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.

– Check the procedures at your installation for handling 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 131 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** *job_name*, *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.
- If you activate DFSMSrmm and reply RETRY, DFSMSrmm issues this message once per system. If you reply CANCEL, DFSMSrmm issues this message for each tape request until you reply RETRY or until you start DFSMSrmm.

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 (**_kilobytes_**K) 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 (**_kilobytes_**K) 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* **U***uuuu*

– 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*, *job_name*, *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.

EDG8122D 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"**

– DFSMSrmm issues this message to prompt the operator to enter a volume into a system-managed library. A volume must be entered into the named library or DFSMSrmm fails the job.

EDG8123D LOGICAL VOLUME *req_volser* **EXPORTED IN STACKED VOLUME** *stack_volser* **LOCATION** *loc_name* **SHELF** *shelf_number* **HOME LOCATION** *home* **- IMPORT VOLUME TO LIBRARY** *lib_name* **AND REPLY** ″**RETRY**″**, OTHERWISE REPLY** ″**CANCEL**″

– DFSMSrmm issues this message to prompt the operator to enter a volume into a system-managed library. Find the volume using the stacked volume name and location information provided in the message. Import the volume into the correct library. If any library is acceptable, you can use the home location name to help determine where the volume should reside.

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.

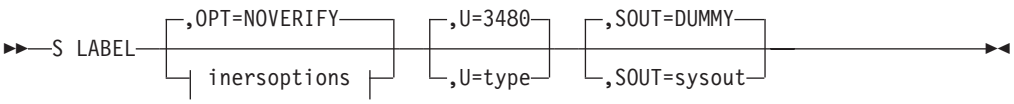
The LABEL procedure is supplied as EDGLABEL in the DFSMSrmm SAMPLIB data set. You must have a tape drive on which you can initialize tape volumes.

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 status of the volume.

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 *z/OS 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 66 on page 150



inersoptions:

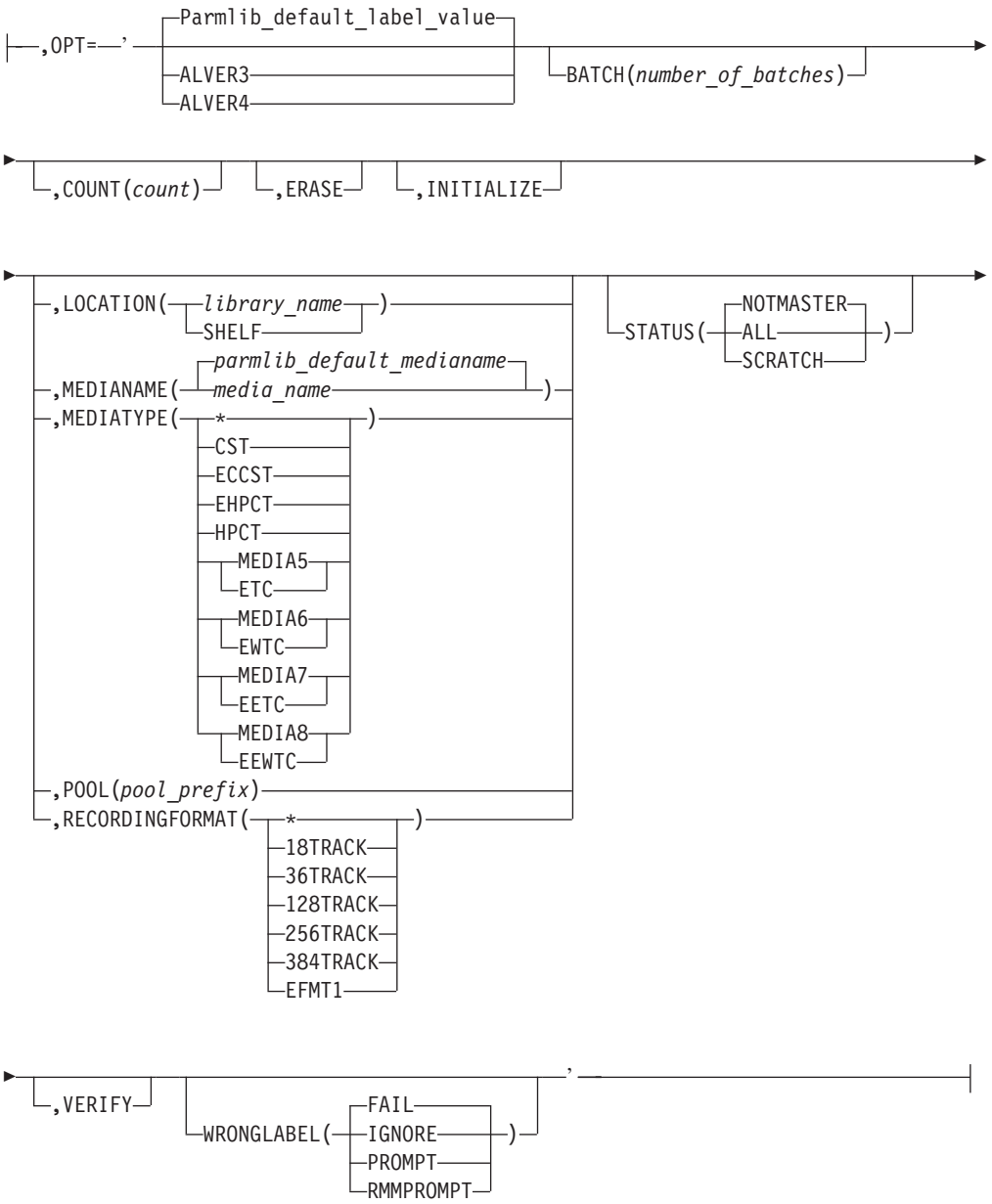


Figure 66. LABEL Command Syntax Diagram

where:

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***|SHELF)**

Use LOCATION to specify a subset of volumes for automatic processing. The *library_name* must be the name of a system-managed tape library that is on the running system. SHELF indicates that the library is a non-system-managed library. If you specify LOCATION, you cannot specify MEDIANAME, MEDIATYPE, POOL, or RECORDINGFORMAT.

There is no default *library_name* value. If you do not specify LOCATION, MEDIANAME, MEDIATYPE, POOL, or RECORDINGFORMAT, DFSMSrmm uses MEDIANAME as the default parameter for automatic processing. This means that DFSMSrmm selects all volumes that are defined with the default media name for processing if they have the required action pending.

MEDIANAME(*parmlib_default_medianame***)**
Use MEDIANAME to specify a subset of volumes for automatic processing. If you specify MEDIANAME, you cannot specify LOCATION, MEDIATYPE, POOL, or RECORDINGFORMAT.

If you do not specify MEDIANAME, MEDIATYPE, LOCATION, POOL, or RECORDINGFORMAT, DFSMSrmm uses MEDIANAME as the default parameter for automatic processing. This means that all volumes that are defined with the default medianame are selected if they have the required action pending.

DFSMSrmm does not use MEDIANAME to set a default media name for the MVS SYSIN INIT and ERASE commands MEDIANAME operand.

The default MEDIANAME is the value you define with the EDGRMMxx parmlib OPTION MEDIANAME operand.

MEDIATYPE(*| CST | ECCST | EHPCT | HPCT | MEDIA5 | MEDIA6 | MEDIA7 |MEDIA8)
Specifies the volume's physical media type. Use MEDIATYPE to specify a subset of volumes for automatic processing. Use one of the following values:

***** The volume is not a cartridge.

CST Cartridge System Tape

ECCST Enhanced Capacity Cartridge System Tape

EHPCT Extended High Performance Cartridge Tape

HPCT High Performance Cartridge Tape

MEDIA5/ETC IBM TotalStorage Enterprise Tape Cartridge

MEDIA6/EWTC
IBM TotalStorage Enterprise WORM Tape Cartridge 3592

MEDIA7/EETC
IBM TotalStorage Enterprise Economy Tape Cartridge 3592

MEDIA8/EEWTC
IBM TotalStorage Enterprise Economy WORM Tape Cartridge 3592

When you specify MEDIATYPE, DFSMSrmm performs automatic processing. If you specify MEDIATYPE, you cannot specify LOCATION, MEDIANAME, POOL, or RECORDINGFORMAT.

There is no default MEDIATYPE value. If you do not specify MEDIATYPE, LOCATION, MEDIANAME, POOL, or RECORDINGFORMAT, DFSMSrmm uses MEDIANAME as the default parameter for automatic processing. This means that all volumes that are defined with the default medianame are selected if they have the required action pending.

Parmlib_default_label_value
DFSMSrmm uses the default label value from the parmlib DEVSUP*xx* member if you do not specify ALVER3 or ALVER4.

POOL(*pool_prefix*)
Use POOL to specify a subset of volumes for automatic processing. A *pool_prefix* value is one-to-five alphanumeric, national, or special characters followed by an asterisk (*). The pool must be one that is defined to DFSMSrmm on the running system. If you specify POOL, you cannot specify LOCATION, MEDIANAME, MEDIATYPE, or RECORDINGFORMAT.

There is no default *pool_prefix* value. If you do not specify POOL, LOCATION, MEDIANAME, MEDIATYPE, or RECORDINGFORMAT, DFSMSrmm uses MEDIANAME as the default parameter for automatic processing. This means that all volumes that are defined with the default medianame are selected if they have the required action pending.

RECORDINGFORMAT(* | 18TRACK | 36TRACK | 128TRACK | 256TRACK | 384TRACK | EFMT1)
Use RECORDINGFORMAT to specify a subset of volumes for automatic processing. RECORDINGFORMAT 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.

18TRACK
Data has been written to the volume in 18-track format. A recording format of 18TRACK is valid with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

36TRACK
Data has been written to the volume in 36-track format. A recording format of 36TRACK is valid with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

128TRACK
Data has been written to the volume in 128-track format. A recording format of 128TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

256TRACK
Data has been written to the volume in 256-track format. A recording format of 256TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

384TRACK
Data has been written to the volume in 384-track format. A recording format of 384TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

EFMT1
Data has been written to the volume in EFMT1 (enterprise format 1) recording format. A recording format of EFMT1 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, and MEDIA8) only.

There is no default RECORDINGFORMAT. If you do not specify RECORDINGFORMAT, LOCATION, MEDIANAME, MEDIATYPE, or POOL, DFSMSrmm uses MEDIANAME as the default parameter for automatic processing. This means that all volumes that are defined with the default media name are selected if they have the required action pending.

STATUS
Use STATUS to control the kind of tapes that you want DFSMSrmm to

initialize or erase. The default for STATUS is NOTMASTER. Specifying STATUS requests automatic processing.

ALL EDGINERS processes all volumes that have the INITIALIZE or ERASE action pending.

NOTMASTER
EDGINERS processes all volumes in SCRATCH, USER, INIT, ENTRY, or PENDING RELEASE status that have the INITIALIZE or ERASE action pending. EDGINERS does not process any volumes in MASTER status. NOTMASTER is the default.

SCRATCH
EDGINERS processes volumes in SCRATCH, INIT, ENTRY, or PENDING RELEASE status that have the INITIALIZE or ERASE action pending. EDGINERS does not process any volumes in MASTER or USER status.

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. Using 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 EDG662E, 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.

RMMPROMPT
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 when you specify 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 67 on page 158 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 vol1_volser, 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 WITH VOL1** *vol1_volser*

– The duplicate 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 *drive_number* **- REQUESTED VOLUME** *volser* **EXPECTED VOL1** *expected_vol1* **MOUNTED VOL1** *mounted_vol1*

– An attempt to relabel a volume was made but the wrong volume, *mounted_vol1*, was mounted instead. 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. Using 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 *drive_number* **- REQUESTED VOLUME** *volser* **EXPECTED VOL1** *expected_vol1* **MOUNTED VOL1** *mounted_vol1* **IS DFSMSrmm MANAGED**

– An attempt was made to relabel a volume but the wrong volume, *mounted_vol1*, was mounted instead. 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 67 on page 158 describes the syntax for replying to prompts for an INIT or ERASE request:

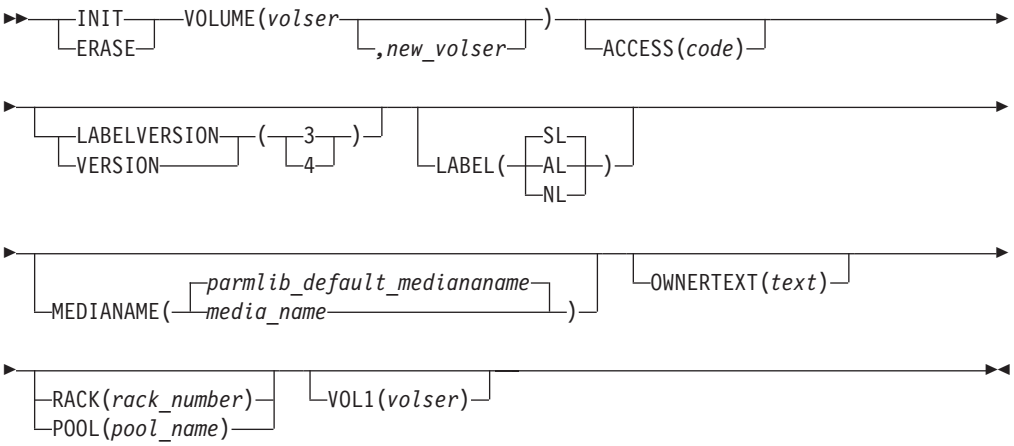


Figure 67. INIT and ERASE Request Syntax Diagram

INIT
Specifies initializing a volume.

You must specify either INIT or ERASE.

ERASE
Specifies a security erase of a volume and writing a new label on the volume.

You must specify 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 you have already defined this new volume in the DFSMSrmm control data set, DFSMSrmm fails the request.

DFSMSrmm adds information about the new volume to the DFSMSrmm control data set. DFSMSrmm 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 z/OS® 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)
Specifies the type of label that should 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.

Specifying LABELVERSION is equivalent to using the CHANGEVOLUME *volser* REQUIREDLABELVERSION subcommand. LABELVERSION 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***|***parmlib_default_medianame***)**
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 that is defined in the control data set. DFSMSrmm fails the request if the media name does not match.

If the volume is not already defined in the control data set and you do not specify a media name, DFSMSrmm uses the parmlib default medianame.

The default MEDIANAME is the value you define with the EDGRMM*xx* parmlib OPTION MEDIANAME operand.

OWNERTEXT(*text***)**
Specifies the owner's name or similar identification. *text* is fourteen characters. Enclose in single quotation marks 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. DFSMSrmm 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.

VOL1(*volser***)**
Specifies a VOL1 label volume serial number to be written in the tape label. To initialize or erase a duplicate volume, specify a value that is different from the volume serial number or the VOL1 label volume serial number that is defined to DFSMSrmm. The *volser* value is one-to-six alphanumeric, national, or special characters.

Example: To initialize a 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 use DFSMSrmm to produce labels by using:

1. A write-to-operator (WTO) on route code 13
2. 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 no OUTPUT JCL statement is provided in the DFSMSrmm started procedure, 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 label 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 68 on page 161 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 69 shows the default label for all other types of volumes. The default label 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  blksiz recf

 volser seqn lab       devc
```

Figure 68. Default Label Format for a Tape Cartridge

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7.
dsname_____________________________________
userdata___________________________________________________________
    jobname_                  crdate____

 dens comp lrecl  blksiz recf expdt_____

               volser seqn lab      devc
```

Figure 69. Default Label Format for a Round Tape

The values for the variables that are shown in Figure 68 and Figure 69 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 Indicator that data on the volume is compacted. 4 characters.

lrecl The logical record length of the data. 1 to 5 characters.

blksiz The block size of the data. 1 to 6 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 9. 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
	ADDVOLUME	AV	Add volume information
	ADDVRS	AS	Add a vital record specification
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
	DELETEVOLUME	DV	Release a volume and delete volume
	DELETEVRS	DS	Delete a vital record specification information
Get	GETVOLUME	GV	Request or assign a volume

© Copyright IBM Corp. 1992, 2004

Table 23. RMM TSO Subcommands (continued)

Group	Subcommand	Abbrev	Function
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
	LISTVOLUME	LV	Display volume information
	LISTVRS	LS	Display vital record specification 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
	SEARCHVOLUME	SV	Create a list of volumes
	SEARCHVRS	SS	Create a list of vital record specifications

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 *z/OS TSO/E User's Guide*.

ADDBIN: Adding a Bin Number in a Storage Location

Before you begin: To use the RMM ADDBIN subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

The ADDBIN subcommand is an alias for the ADDRACK subcommand. See "ADDRACK: Adding a Shelf Location" on page 182 for the combined description of the ADDRACK and ADDBIN operands.

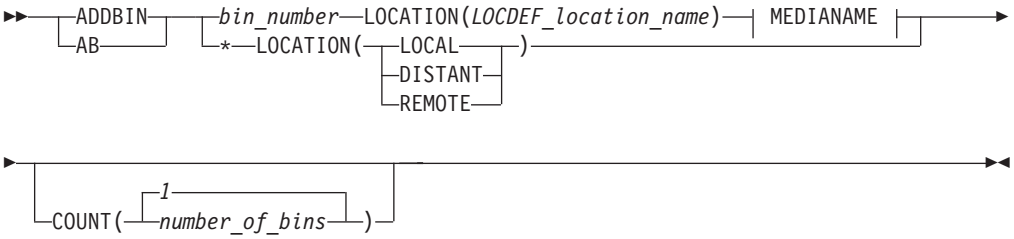
DFSMSrmm defines shelf space in storage locations as bin numbers. Use the ADDBIN subcommand to define empty or available shelf locations in storage locations. The storage locations can be DFSMSrmm built-in storage locations or storage locations that are defined by your installation.

To add one or more bin numbers to an installation-defined storage location, 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, select an initial bin number that is long enough and low enough to accommodate the count value you specify, 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 CNTL subcommand to determine how many bin numbers are 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 287 for more information.

Syntax



MEDIANAME:

Operands

*bin_number|**

Specifies the shelf location in a storage location. 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 the number of bins 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.

Examples

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)
```

Task: Add ten empty bin numbers to the LOCAL built-in storage location.

Command:

```
RMM ADDBIN * LOCATION(LOCAL) COUNT(10)
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin:

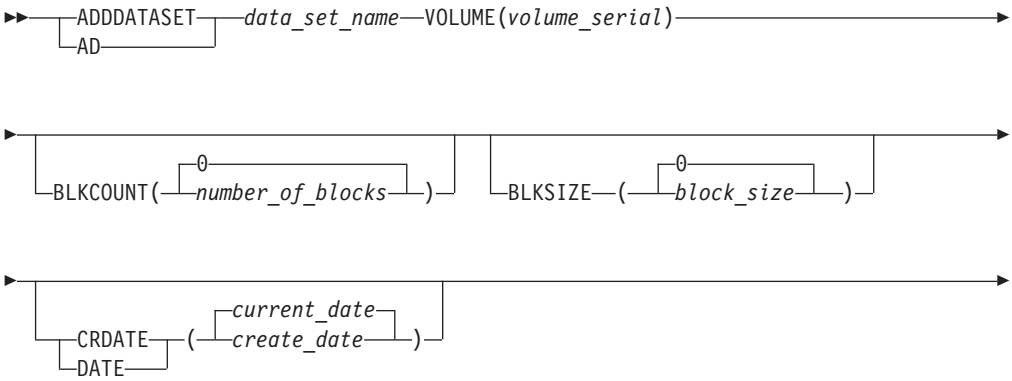
- To use the RMM ADDDATASET subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.
- To use the RMM ADDDATASET FORCE subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile and UPDATE access to the STGADMIN.EDG.FORCE resource profile to add a data set to a volume where information was recorded during O/C/EOV processing.
- Define information about the volume where the data set resides to DFSMSrmm before you add information about the data set. Use the ADDVOLUME subcommand to define volumes to DFSMSrmm. See “ADDVOLUME: Adding Volume Information” on page 186 for more information.
- Define information about data sets preceding the data set you want to add to DFSMSrmm, or DFSMSrmm fails your requests. 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.

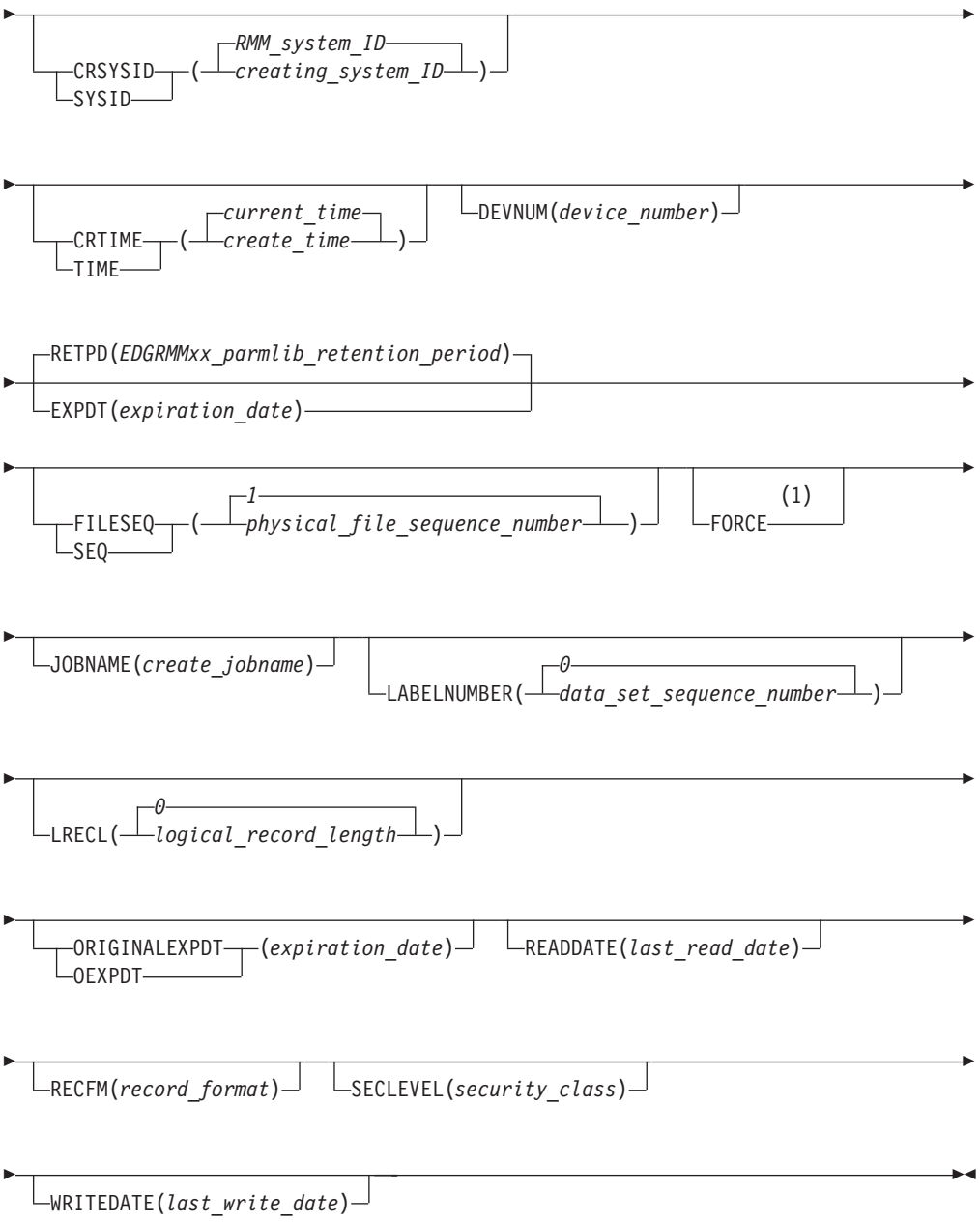
Use the ADDDATASET subcommand to manually define a data set on a volume to DFSMSrmm. The volume on which the data set resides must have either master or user status.

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, DFSMSrmm uses the data set name to retain a data set when you do not define a job name for the data set.

Syntax





Notes:

1 Use the FORCE operand, which requires CONTROL access to the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource, to add a data set to a volume that was recorded by DFSMSrmm during O/C/EOV processing,.

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 999999. 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 usage.

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. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required. You can specify a date in the range between 0000/000 to 9799/365.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

You can specify a date in the range between 00000 to 99366.

The default is the date when you issue the ADDDATASET subcommand.

CRSYSID/SYSID(*creating_system_ID***)**
Specifies the ID of the system on which the data set was created. Use a one to eight character unique system name. Default: RMM_system_ID.

When you run DFSMSrmm with unshared catalogs, DFSMSrmm uses the CRSYSID of the first file on a volume to determine the system where the volume should return to scratch.

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.

You can specify a time in the range between 000000 to 235959.

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.

EXPDT(*expiration_date***)**
Specifies the date when the data set should be considered for release. The expiration date cannot exceed the maximum retention period (MAXRETPD) set by your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

Supply the year and day in one of two forms. We recommend that you use the *yyyy/ddd* format rather than the *yyddd* format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

EXPDT is mutually exclusive with RETPD. If you do not specify EXPDT or RETPD, DFSMSrmm calculates the expiration date using the DFSMSrmm EDGRMM*xx* parmlib member RETPD value.

FILESEQ(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 65535. 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
Specifies overriding 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-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 65535. A value of 0 indicates that the data set sequence is unknown. The value that you specify is not validated with the values known for either the preceding or following data sets that are 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 99999.

The default value is 0.

ORIGINALEXPDT/OEXPDT(*expiration_date***)**
Specifies the original JCL expiration date of the data set.

Supply the year and day in one of two forms. We recommend that you use the *yyyy/ddd* format rather than the *yyddd* format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

There is no default.

READDATE(*last_read_date***)**
Specifies when the data set was last read.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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.

RETPD(*retention_period***)**
Specifies the number of days that DFSMSrmm retains the data set before considering it for release. *retention_period* is a one-to-four character decimal number. The retention period is added to today's date to create the expiration date. The expiration date cannot exceed the maximum retention period (MAXRETPD) specified for your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

RETPD is mutually exclusive with EXPDT.

If you do not specify RETPD or EXPDT, DFSMSrmm uses the default retention period from the parmlib member EDGRMM*xx*.

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 287 for more information. See *z/OS 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 parmlib 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 number is one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. This operand is required.

WRITEDATE(*last_write_date***)**
Specifies when the data set was last written to tape.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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.

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)
```

Return Codes

See Chapter 10, "DFSMSrmm Return Codes and Reason Codes," on page 355 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. The data is incomplete or incorrect 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

Before you begin: To use the RMM ADDOWNER subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

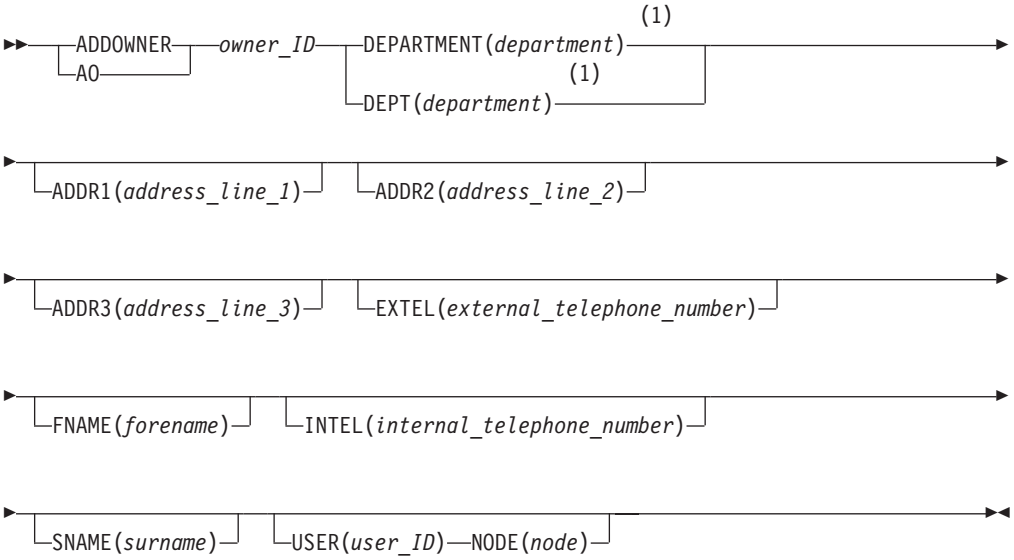
Use the ADDOWNER subcommand to define an owner to DFSMSrmm. An owner can be an individual or a group defined by a RACF group name, or any other value you choose.

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 operand and the NODE operand. 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 186 or "CHANGEVOLUME: Changing Volume Information" on page 234 for more information. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information on setting parmlib options.

Syntax



Notes:

1 The DEPARTMENT operand must contain at least one non-blank character.

Operands

ADDR1(*address_line_1***)**
Specifies the first address line. An address line is one to forty characters enclosed in single quotation marks 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 quotation marks 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 quotation marks 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 quotation marks 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 quotation marks 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 quotation marks 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 quotation marks if it contains any special characters or blanks. The default is blanks.

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 quotation marks 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.

Example

Task: Add details for the new owner information shown in Table 24.

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)
```

Return Codes

See Chapter 10, "DFSMSrmm Return Codes and Reason Codes," on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDPRODUCT: Adding Software Product Information

Before you begin: To use the RMM ADDPRODUCT subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

Use the ADDPRODUCT subcommand to define a software product to DFSMSrmm. You must enter the product number and name when adding a software product to DFSMSrmm.

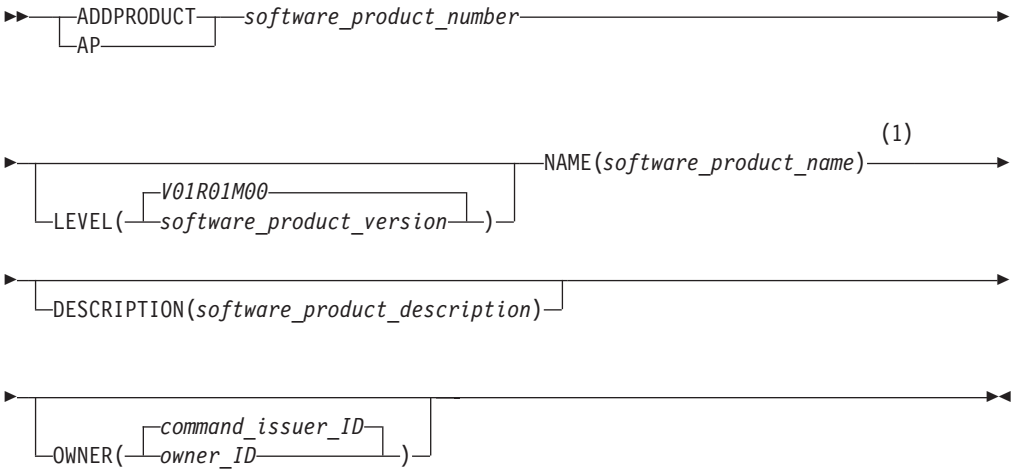
After you have defined the software product to DFSMSrmm, use the ADDVOLUME subcommand to associate one or more volumes with the software product. You must define the software product to DFSMSrmm before you associate one or more volumes with it. The volumes must not be already defined to DFSMSrmm. Use the ADDVOLUME subcommand to define new volumes. See “ADDVOLUME: Adding Volume Information” on page 186 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, 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 29 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 232 for more information.

If your installation has set up the NOTIFY option, DFSMSrmm notifies the designated product owner when a program product volume is added. The owner record must contain a valid user ID and node ID.

Syntax



Notes:

1 The NAME operand must contain at least one non-blank character.

Operands

DESCRIPTION(*software_product_description***)**
Specifies descriptive text about the software product. The descriptive text is one to thirty characters enclosed in single quotation marks 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 quotation marks 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 322 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 quotation marks if it contains any special characters or blanks. This operand is required and must immediately follow the ADDPRODUCT subcommand.

Example

Task: Add details of the z/OS software product with product number 5694-A01, version 1.1.0.

Command:

```
RMM ADDPRODUCT '5694-A01' NAME('z/OS') LEVEL(V01R1M00)
```

Return Codes

See Chapter 10, "DFSMSrmm Return Codes and Reason Codes," on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

ADDRACK: Adding a Shelf Location

Before you begin: To use the RMM ADDRACK subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

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 166 for information about using the RMM ADDRACK subcommand alias.

Use the ADDRACK subcommand 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 available rack numbers 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 *z/OS 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, use an initial rack number that can accommodate the number of volumes you might add. For example, if you supply an initial rack number of RA9992, you can only add eight rack numbers. If you add ten rack numbers, 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 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.

Table 25 shows the relationship between rack numbers and the three types of volumes that DFSMSrmm manages.

Table 25. Assigning Rack Numbers to Volumes that DFSMSrmm Manages

For a	You	Optionally
Physical Volume	Can specify a rack number or pool prefix.	DFSMSrmm tries to use the volume serial number of the volume as the rack number. If DFSMSrmm cannot find the rack number that matches the volume serial number, no rack number is assigned.
Logical Volume	Cannot specify a rack number or pool prefix.	DFSMSrmm does not assign a rack number to the logical volume because the logical volume is associated with a stacked volume container or library rather than a rack number or pool.

Table 25. Assigning Rack Numbers to Volumes that DFSMSrmm Manages (continued)

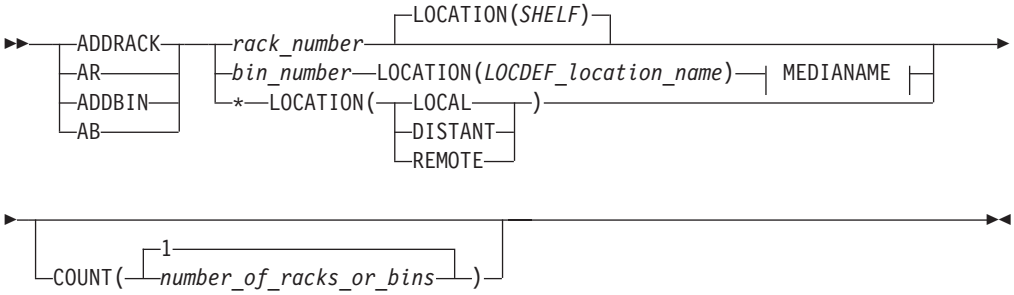
For a	You	Optionally
Stacked Volume	Can specify a rack number or pool prefix.	DFSMSrmm tries to use the volume serial number of the volume as the rack number. If DFSMSrmm cannot find the rack number that matches the volume serial number, no rack number is assigned.

To add bin numbers to a built-in storage location, use * as the bin number and provide a storage location name. You can also specify 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 specify how many bins you want to add.

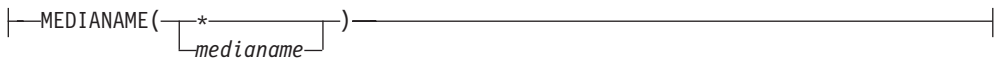
Use the LISTCONTROL subcommand with the CNTL operand to determine how many bin numbers are already defined and how many bin numbers are currently in use as shown in “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 287.

See “LISTBIN: Displaying Information about a Shelf Location” on page 285 to obtain information about individual bin numbers.

Syntax



MEDIANAME:



Operands

*bin_number,**

Specifies the shelf location in a storage location. 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 the number of rack numbers to add to the removable media library or the number of 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. A removable media library is either a SHELF location or a system-managed library.

The DFSMSrmm built-in storage location names are LOCAL, DISTANT, and REMOTE. You cannot use the 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 specified. If you do not use the LOCATION operand, DFSMSrmm adds rack numbers to the removable media library.

MEDIANAME(*medianame***|*)**
Specifies 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 must contain numeric suffixes so that DFSMSrmm can automatically define each new rack number. A rack number is one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters.

You must use either a bin number, rack number or an asterisk (*), immediately following the ADDRACK subcommand.

If you specify *library_name* as the LOCATION value, the rack number must be the same as the volume serial number.

Examples

Task: Add five empty racks to the removable media library, starting with rack number ABUB02.

Command:

```
RMM ADDRACK ABUB02 COUNT(5)
```

Task: Add ten empty bin numbers to the LOCAL storage location by using the alias ADDBIN.

Command:

```
RMM ADDBIN * LOCATION(LOCAL) COUNT(10)
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin: To use the RMM ADDVOLUME subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

Use the ADDVOLUME subcommand 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 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.

When you are adding volumes that reside in a manual tape library, you must specify the MEDIATYPE operand.

You can use the RMM ADDVOLUME subcommand to define volumes to DFSMSrmm. When you are defining WORM tapes to DFSMSrmm, allow DFSMSrmm to record the volume WWID when the volume is first used on the system rather than specifying the WWID yourself. This ensures that the WWID for the volume is recorded correctly.

When you add a volume to DFSMSrmm, you can also specify a rack number or a pool prefix for a physical or stacked volume. You cannot specify a rack number or pool prefix for a logical volume. If you do not specify a rack number or pool prefix, DFSMSrmm tries to allocate one for the volume as described in Table 26.

Table 26. How DFSMSrmm Assigns Rack Numbers to a Volume

For a	If you do not specify a rack number or a pool prefix
Physical volume	DFSMSrmm tries to use the volume serial number of the volume as the rack number. If DFSMSrmm cannot find the rack number that matches the volume serial number, no rack number is assigned.
Logical volume	DFSMSrmm does not assign a rack number to the logical volume because the logical volume's association is with a stacked volume container or library rather than a rack number or pool.
Stacked volume	DFSMSrmm tries to use the volume serial number of the volume as the rack number. If DFSMSrmm cannot find the rack number that matches the volume serial number, no rack number is assigned.

Use the TYPE operand to indicate the type of volume that is being added. When you add a volume that resides in a system-managed library, you do not have to specify the TYPE operand because DFSMSrmm can identify the volume type. If you are adding a stacked volume or a logical volume to a location other than a system-managed VTS, we recommend that you always specify the TYPE operand.

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.

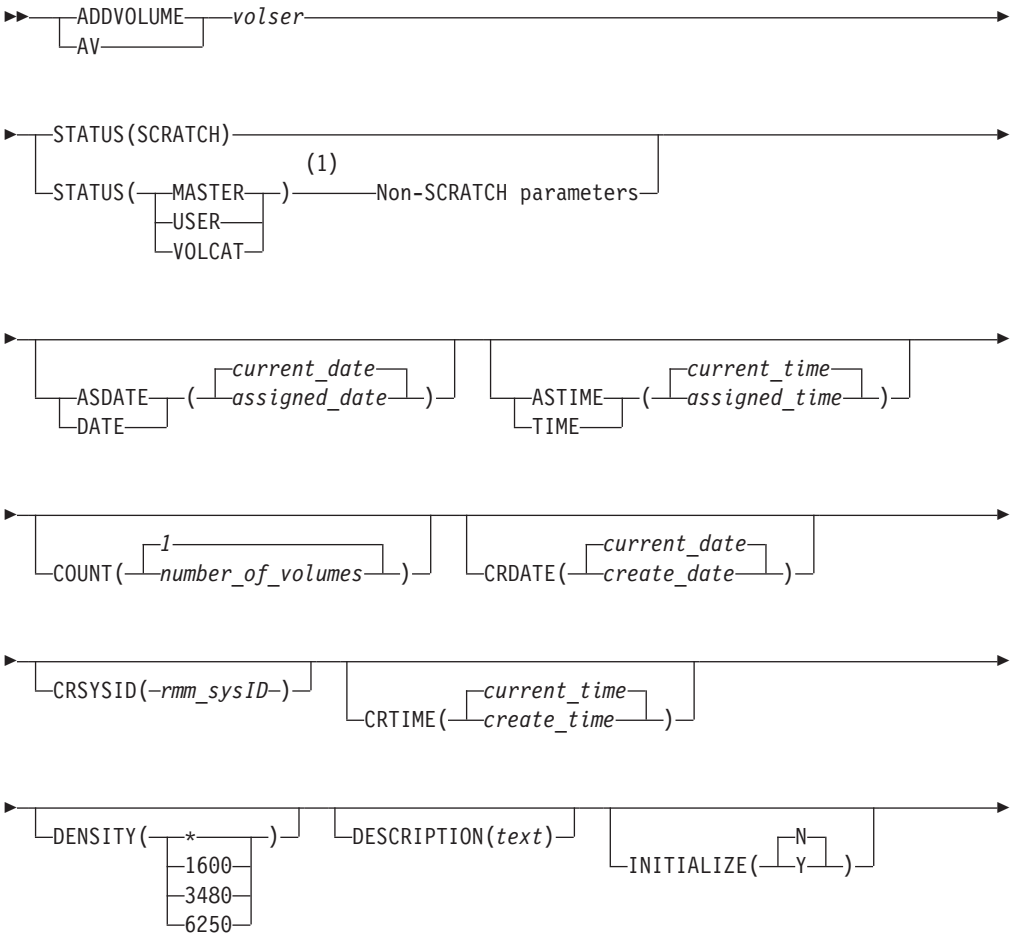
Use the NUMBER, FEATCD, and LEVEL operands to associate the volume with a software product. The software product must be already defined to DFSMSrmm before you can associate it with a volume.

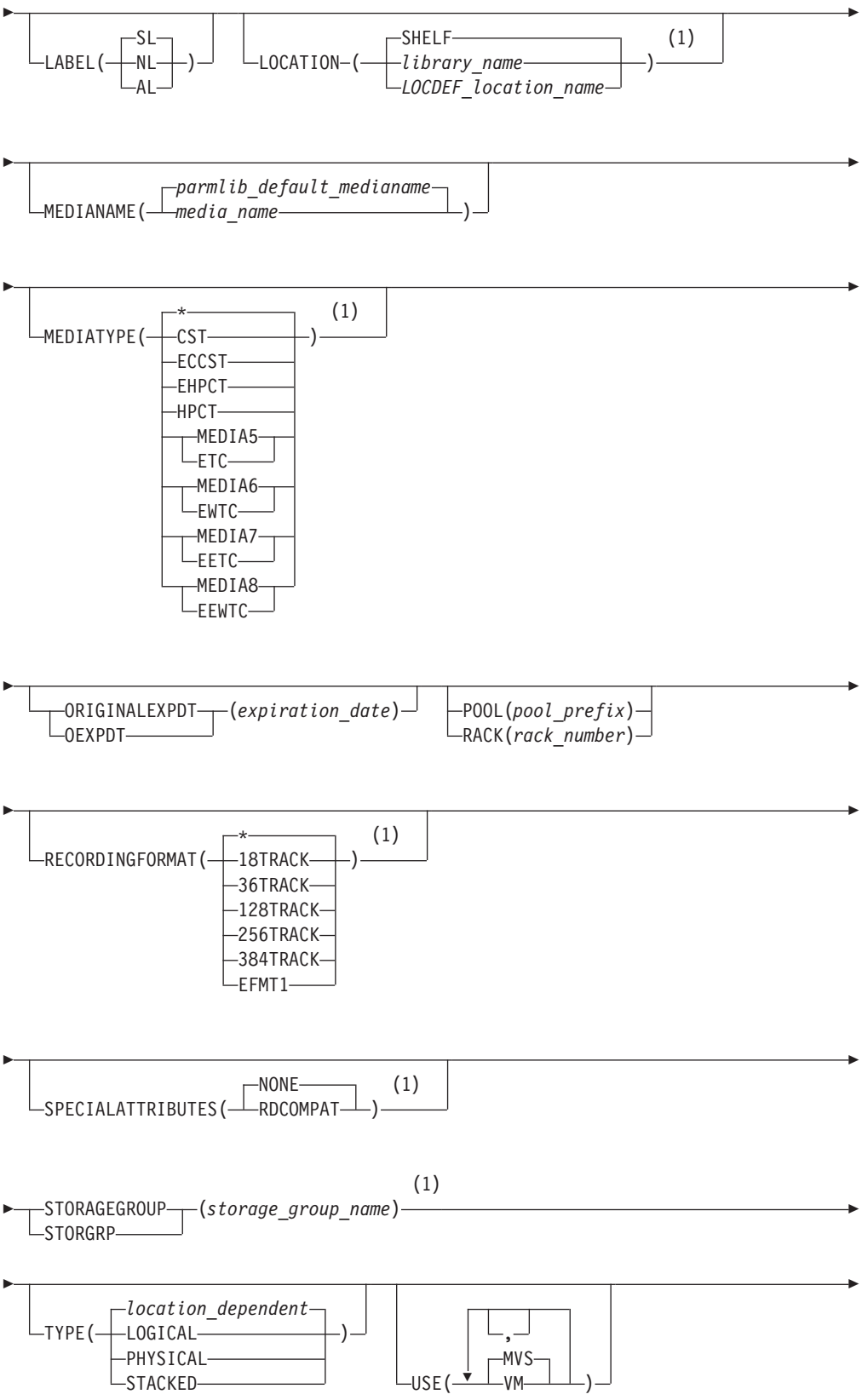
When you specify ADDVOLUME STATUS(VOLCAT), DFSMSrmm uses information in the TCDB to update the DFSMSrmm control data set. The control data set information that can be updated includes: COMPACTION, CONTAINER, EXPDT, HOME, LOCATION, MEDIATYPE, READDATE, RECORDINGFORMAT, SPECIALATTRIBUTES, STATUS, STORAGEGROUP, TYPE, and WRITEDATE.

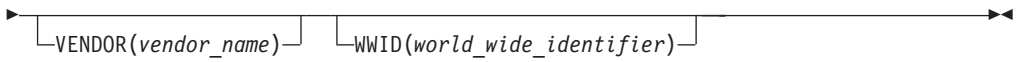
If you use any non-scratch operands when adding scratch volumes, DFSMSrmm ignores the non-scratch operands.

Syntax

ADDVOLUME Volume Operands




LABEL(SL NL AL)
LOCATION (SHELF library_name LOCDEF_location_name) (1)
MEDIANAME(parmlib_default_medianame media_name)
MEDIATYPE(* CST ECCST EHPCT HPCT MEDIA5 ETC MEDIA6 EWTC MEDIA7 EETC MEDIA8 EEWTC) (1)
ORIGINALEXPDT (expiration_date)
OEXPDT
POOL(pool_prefix)
RACK(rack_number)
RECORDINGFORMAT(* 18TRACK 36TRACK 128TRACK 256TRACK 384TRACK EFMT1) (1)
SPECIALATTRIBUTES(NONE RDCOMPAT) (1)
STORAGEGROUP (storage_group_name) (1)
STORGRP
TYPE(location_dependent LOGICAL PHYSICAL STACKED)
USE(, MVS VM)

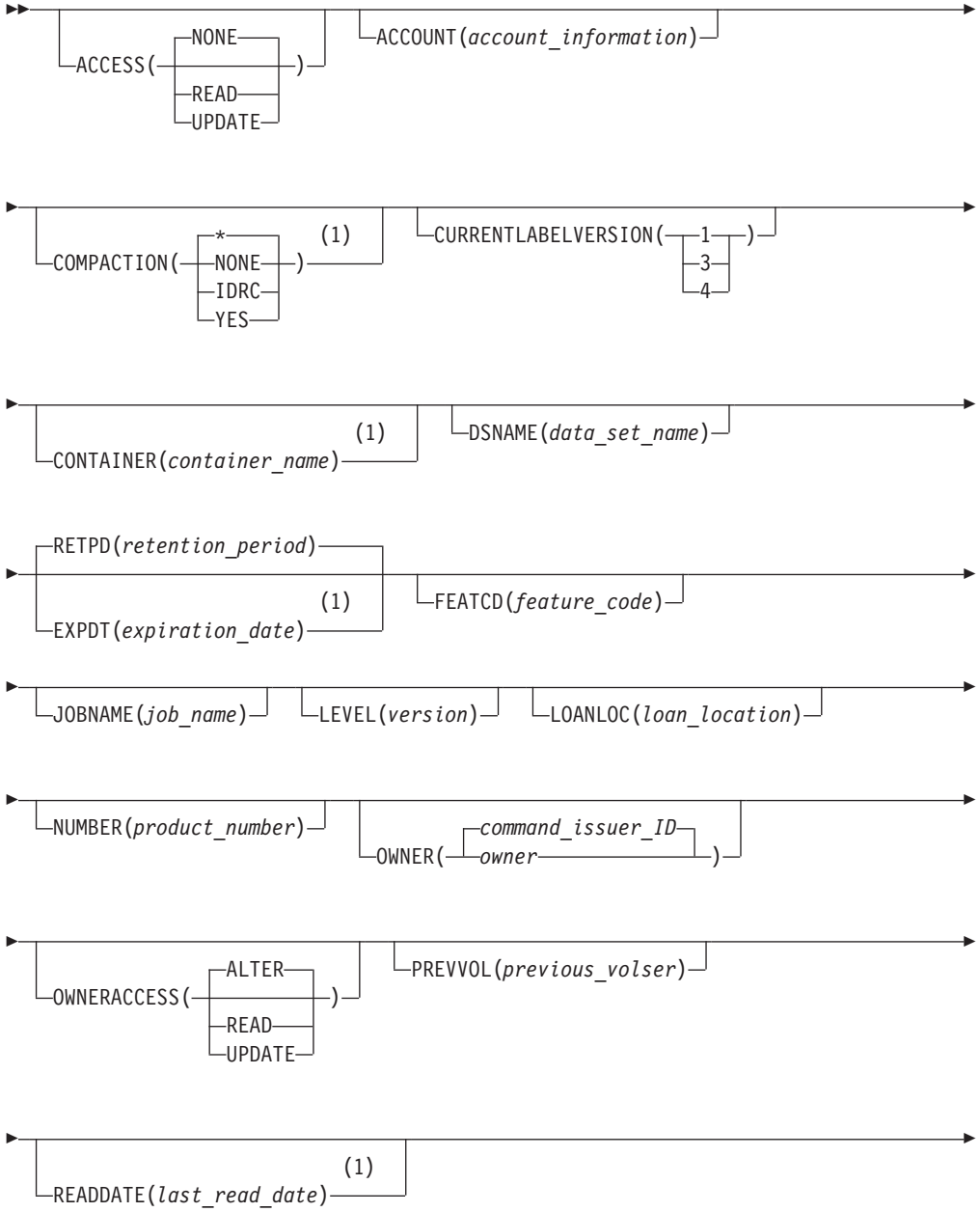


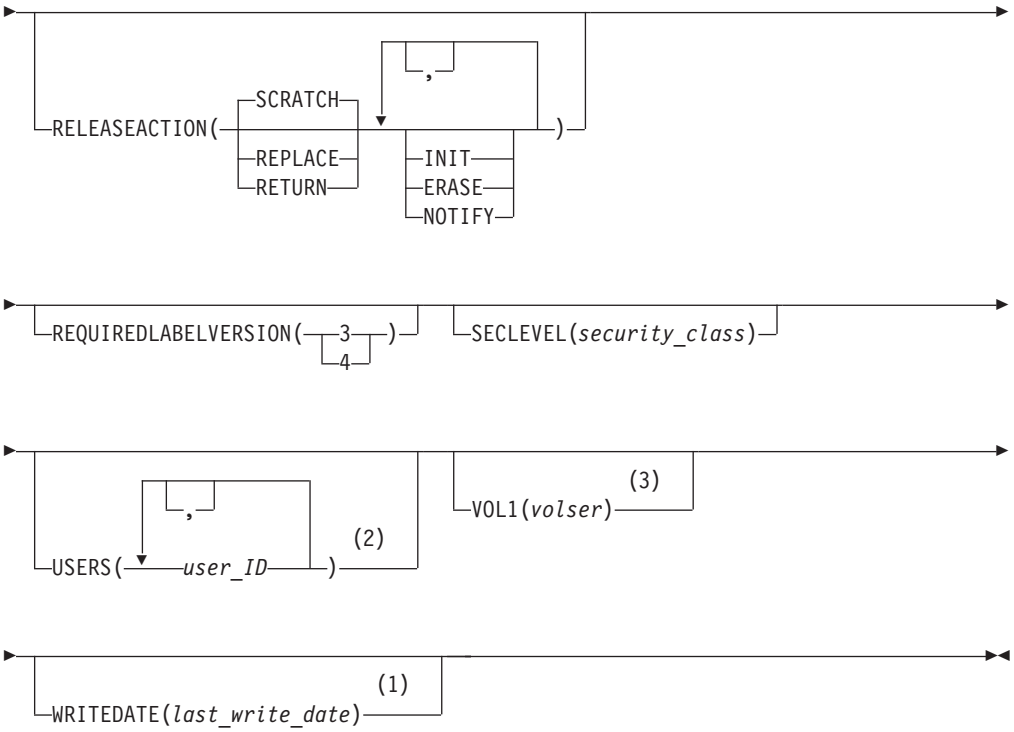
Notes:

1 This operand uses the tape configuration database information when STATUS(VOLCAT) is specified.

Syntax

ADDVOLUME Non-scratch Volume Optional Operands





Notes:

1 This operand uses the tape configuration database information when STATUS(VOLCAT) is specified.

2 You can specify a maximum of 12 user IDs.

3 This operand cannot be specified for SCRATCH volumes, LOGICAL volumes, STACKED volumes, or NOLABEL volumes.

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 quotation marks if it contains any special characters or blanks.

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).

There is no default.

ASDATE(*assigned_date***)**
Specifies the date when a master volume or user volume was assigned. 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. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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.

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.

The default is *.

CONTAINER(*container_name***)**
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.

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.

CRDATE(*create_date* **|** *current_date***)**
Specifies the date when the volume was created.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required. You can specify a date in the range between 0000/000 to 9799/365.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

You can specify a date in the range between 00000 to 99366.

The default is the date you issue the ADDVOLUME subcommand.

CRSYSID(*RMM_sysID***)**
Specifies the ID of the system on which the volume was created. Specify a unique system name one-to-eight characters long.

The default value is the DFSMSrmm OPTION SYSID value.

CRTIME(*create_time* **|** *current_date***)**

Specifies the time when the volume was created. The CRTIME format is *hhmmss* where:
- *hh* is hours
- *mm* is minutes
- *ss* is seconds

For example, nine o'clock in the morning is 090000.

You can specify a time in the range between 000000 to 235959.

The default is the time when you issue the ADDVOLUME subcommand.

CURRENTLABELVERSION(1|3|4)
Specifies the ISO/ANSI label version for the volume that you are adding.

There is no default.

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.

There is no default.

DESCRIPTION(*text*)
Specifies descriptive text about the volume. Descriptive text is one to thirty characters enclosed in single quotation marks 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).

There is no default.

EXPDT(*expiration_date*)
Specifies the date the volume should be considered for release. The expiration date cannot exceed the maximum retention period (MAXRETPD) set by your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

To use the dates 99365 and 99366 which mean permanent retention, you must specify the MAXRETPD NOLIMIT operand in the DFSMSrmm EDGRMMxx parmlib member.

EXPDT is mutually exclusive with RETPD.

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.

If you do not specify EXPDT or RETPD, DFSMSrmm uses the default retention defined in the DFSMSrmm EDGRMM*xx* parmlib member.

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).

There is no default.

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. INITIALIZE(Y) is not supported for a stacked volume.

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, 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).

There is no default.

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 304 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.

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 quotation marks 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).

There is no default.

LOCATION(SHELF|*library_name***|***LOCDEF_location_name***)**
Specifies the location where the volume is stored and sets the home location for the volume. Use one of the following values.

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, or an automated tape library such as the IBM TotalStorage Enterprise Automated Tape Library (3495). 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, DFSMSrmm adds the volume to the TCDB; if the library is an automated tape library 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. DFSMSrmm does not add the volume to the TCDB since the system does this when the volume enters the automated tape library. You cannot specify a distributed library name.

If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

LOCDEF_location_name
Indicates that the volume is stored in a storage location. When you store volumes in a storage location as their home location, you enable volumes to return to scratch in the named storage location. Storing volumes in this way allows you to avoid using the location SHELF.

When you use a shelf-managed storage location as a home location, DFSMSrmm sets the required location to the specified location. DFSMSrmm sets the current location to SHELF and enables the next inventory management DSTORE run to assign an available bin number.

The default is SHELF.

MEDIANAME(*parmlib_default_medianame***|***media_name***)**
Specifies the physical shape of a volume or the type of a volume. Specify a one-to-eight character name. To help you identify the media, you can use media names that describe size or shape. Using size or shape to describe the media rather than using a specific media name, gives you more flexibility in the media that can reside in a pool. Some examples of MEDIANAME that you might define include: CART, ROUND, SQUARE, 3420, 3480, TAPE, OPTICAL, and CASSETTE.

You can use any name for a MEDIANAME because DFSMSrmm does not check that the media name is a device type that has been defined to MVS. Use MEDIANAME to identify different types of physical shelf space for different

media or to distinguish different media characteristics such as cartridge tape and enhanced capacity cartridge system tape.

The *parmlib_default_medianame* is set in the DFSMSrmm EDGRMM*xx* parmlib member. For more information on how to set this value, see *z/OS DFSMSrmm Implementation and Customization Guide*.

The default value is *parmlib_default_medianame*.

MEDIATYPE(*|CST|ECCST|EHPCT|HPCT|MEDIA5|MEDIA6|MEDIA7|MEDIA8)
Specifies the volume's physical media type. Use one of the following:

*****	The volume is not a cartridge.
CST	Cartridge System Tape
ECCST	Enhanced Capacity Cartridge System Tape
EHPCT	Extended High Performance Cartridge Tape
HPCT	High Performance Cartridge Tape
MEDIA5/ETC	IBM TotalStorage Enterprise Tape Cartridge
MEDIA6/EWTC	IBM TotalStorage Enterprise WORM Tape Cartridge 3592
MEDIA7/EETC	IBM TotalStorage Enterprise Economy Tape Cartridge 3592
MEDIA8/EEWTC	IBM TotalStorage Enterprise Economy WORM Tape Cartridge 3592

You must specify the MEDIATYPE operand when you are adding volumes that reside in a manual tape library.

If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB.

See Table 6 on page 27 for information about how DFSMSrmm assigns MEDIATYPE and RECORDINGFORMAT.

The default value is *.

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 quotation marks 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).

There is no default.

ORIGINALEXPDT/OEXPDT(*expiration_date*)
Specifies the original JCL expiration date of the volume. It should be the highest original expiration date of all the files on the volume.

Supply the year and day in one of two forms. We recommend that you use the *yyyy/ddd* format rather than the *yyddd* format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.

- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

There is no default.

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.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

The default is the user ID of the command issuer.

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, national, or special characters followed by an asterisk. Enclose it in quotation marks if it contains any special characters.

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 287 for more information.

If you do not supply either a pool prefix or a rack number, DFSMSrmm assigns the volume a rack number as described in Table 26 on page 186. 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. You cannot specify POOL when TYPE(LOGICAL) is specified. POOL cannot be used with RACK.

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, national, or special characters. Enclose it in single quotation marks if it contains any special characters.

Restriction: 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 in their correct sequence.

There is no default.

RACK(*rack_number***)**
Specifies a shelf location in the removable media library where DFSMSrmm stores the volume. A rack number is one-to-six alphanumeric, national, or

special characters. Enclose it in single quotation marks if it contains any special characters. The rack number that you use must be previously defined and must be empty.

If you do not supply a pool prefix or a rack number, DFSMSrmm assigns the volume a rack number as described in Table 26 on page 186. RACK cannot be used 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. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

There is no default.

This operand is ignored if you use STATUS(SCRATCH). If you use STATUS(VOLCAT), DFSMSrmm overrides the value you specify with information from the TCDB.

RECORDINGFORMAT(* | 18TRACK | 36TRACK | 128TRACK | 256TRACK | 384TRACK | EFMT1)
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.

18TRACK
Data has been written to the volume in 18-track format. A recording format of 18TRACK is valid with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

36TRACK
Data has been written to the volume in 36-track format. A recording format of 36TRACK is valid with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

128TRACK
Data has been written to the volume in 128-track format. A recording format of 128TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

256TRACK
Data has been written to the volume in 256-track format. A recording format of 256TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

384TRACK
Data has been written to the volume in 384-track format. A recording format of 384TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

EFMT1
Data has been written to the volume in Enterprise Format 1 recording technology. You can only specify EFMT1 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), and MEDIATYPE(MEDIA8).

Recommendation: Specify the known recording format for system-managed, non-scratch volumes. If you use an incorrect value, the volume can be mounted on a tape drive that can neither read nor write to the volume. If you do not specify a value, DFSMSrmm sets a default value that is valid for the media type you specify.

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.

See Table 6 on page 27 for information about how DFSMSrmm assigns MEDIATYPE and RECORDINGFORMAT.

The default value is *.

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. 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(0|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. Specify 0 when you have no required label version.

The default is 0.

RETPD(*retention_period*)
Specifies the number of days that DFSMSrmm retains the volume before considering it for release. *retention_period* is a one-to-four decimal value. The retention period is added to today's date to create the expiration date. The expiration date cannot exceed the maximum retention period (MAXRETPD) set by your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

RETPD is mutually exclusive with EXPDT.

If you do not specify RETPD or EXPDT, DFSMSrmm uses the default retention period from the parmlib member EDGRMM*xx*.

SECLEVEL(*security_class*)
Specifies the volume's security class. This value is one to eight characters, and must be one defined for your installation.

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 287 for more information.

This operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

There is no default.

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.

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.

There is no default.

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. A storage group name can be a value that matches to a VLPOOL NAME value but does not need to be defined on a VLPOOL definition.

For volumes in a system-managed library, DFSMSrmm uses the current location or the location specified on the command to validate the specified storage group. For volumes with a system-managed home location defined, DFSMSrmm uses the home location for validation. For other volumes, any value you specify is accepted as long as the value is valid in the current SMS configuration.

You can set the storage group even if the TCDB already contains a storage group name.

A storage group name can be assigned to any volume, even a scratch volume. The storage group name can be used for scratch pooling, except when the volume is in a system-managed automated tape library. For system-managed scratch volumes, the storage group name is not maintained in the TCDB because it is not supported by SMS tape processing.

When you do not specify a storage group name, DFSMSrmm assigns a storage group name by using the matching EDGRMMxx VLPOOL NAME value. If the VLPOOL NAME value is a valid storage group name, DFSMSrmm uses the VLPOOL NAME value as the default value for volumes added to the pool.

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|STACKED)
Specifies the volume type. When you do not specify a volume type, DFSMSrmm sets the volume type based on the location where the volume resides. TYPE(PHYSICAL) can be used for volumes that do not reside in a VTS. TYPE(LOGICAL) and TYPE(STACKED) can also be specified for volumes that reside outside a VTS. TYPE(LOGICAL) and TYPE(STACKED) cannot be specified for volumes in a system-managed tape library that is not a VTS. DFSMSrmm sets the default volume type to TYPE(PHYSICAL) for volumes that do not reside in a VTS. DFSMSrmm sets the default volume type for volumes that reside in a VTS as TYPE(STACKED) when the volume is known to the VTS as a stacked volume. DFSMSrmm sets the default volume type to TYPE(LOGICAL) for any volumes in a VTS that are not stacked volumes.

When you specify TYPE(STACKED) or DFSMSrmm sets the default to stacked, DFSMSrmm forces the volume status to master status. DFSMSrmm also sets the following values for stacked volumes.

- Assign date and time each time a stacked volume becomes empty and each time the first volume is added into it as a container.

- Last read date each time a volume is imported or removed from the stacked volume.
- Last write date each time a volume is exported or added to the stacked volume.
- Security level each time the stacked volume has the first volume added. Security level is set to the added volume's security level. Each time a volume is added to a stacked volume, DFSMSrmm sets the stacked volume's security level to the maximum of the existing and new volume security level.
- Use count each time the stacked volume has the first volume added. Use count is reset each time a new export is started and incremented each time a volume is added to the stacked volume.
- Usage which is the sum of all the volumes added to the stacked volume. Usage is reset each time a new export is started and incremented each time a volume is added to the stacked volume.

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).

There is no default.

VENDOR(*vendor_name***)**
Specifies the manufacturer or supplier of the volume. The vendor value is 1 to 8 alphanumeric or special characters that you can set or change at any time. DFSMSrmm never changes or uses this value. You can use it for reporting purposes and tracking supplier information for batches of volumes.

There is no default.

volser
Specifies the volume serial number. A volume serial number is one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. When you add volumes into a system-managed library, you can only specify volume serial numbers consisting of alphanumeric characters. You cannot use a generic volume serial number.

volser is required and must follow the ADDVOLUME subcommand.

VOL1(*volser***)**
Specifies a standard label volume serial number. Use this operand to define a duplicate volume when the VOL1 label volume serial number does not match the volume you are defining to DFSMSrmm. The variable *volser* is one-to-six alphanumeric, national, or special characters.

There is no default.

WRITEDATE(*last_write_date***)**
Specifies when the volume was last written to.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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.

WWID(*world_wide_identifier***)**
Specifies the unique world wide identifier set on the volume by the manufacturer. The world wide ID is 12 hexadecimal characters. You can specify 24 characters 0-9 and A-F.

In some publications, the world-wide unique cartridge identifier (WWCID) may also be referred to as the world-wide identifier (WWID), as in this document.

There is no default.

Examples

Task: Add a duplicate volume to the removable media library using a VOL1 value A01234.

Command:

```
RMM ADDVOLUME B12345 COUNT(1) STATUS(MASTER) MEDIANAME(3480) -
       OWNER(RMMUSER) VOL1(A01234)
```

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 already been defined. The volumes must be labeled before they can be used, as indicated by INIT(Y) operand.

Command:

```
RMM ADDVOLUME S00000 COUNT(1000) STATUS(SCRATCH) INIT(Y)
```

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)
```

Because MVS is a default value, you can enter the following command.

```
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)
```

Task: Add a WORM volume.

Command:

```
RMM ADDVOLUME MW0001 STATUS(SCRATCH) VENDOR('batch 33')
    WWID(x'1A2B3C4DA1B27E8F90E0F000')
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin: To use the RMM ADDVRS subcommand, you need CONTROL access to the STGADMIN.EDG.VRS resource profile to add vital record specifications.

Use the ADDVRS subcommand 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 4, "Retaining and Moving Your Volumes," on page 55 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 *z/OS DFSMSrmm Implementation and Customization Guide* for more information.

You must run vital record processing when you add or change a vital record specification for DFSMSrmm to apply the policy defined by the vital record specification. You should reclaim any volumes that are pending release or ready to return to scratch to avoid data loss. Use the RMM CHANGEVOLUME subcommand to change the status of these volumes to reclaim them. 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 86 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 the location priority you define in a vital record specification, the default priority, or one your installation defines using the DFSMSrmm parmlib member EDGRMMxx LOCDEF command. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information.

Data Set Vital Record Specifications

You can define vital record specifications for data sets or groups of data sets using the ADDVRS with the DSNAME operand and either specific data set names or data set name masks. 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

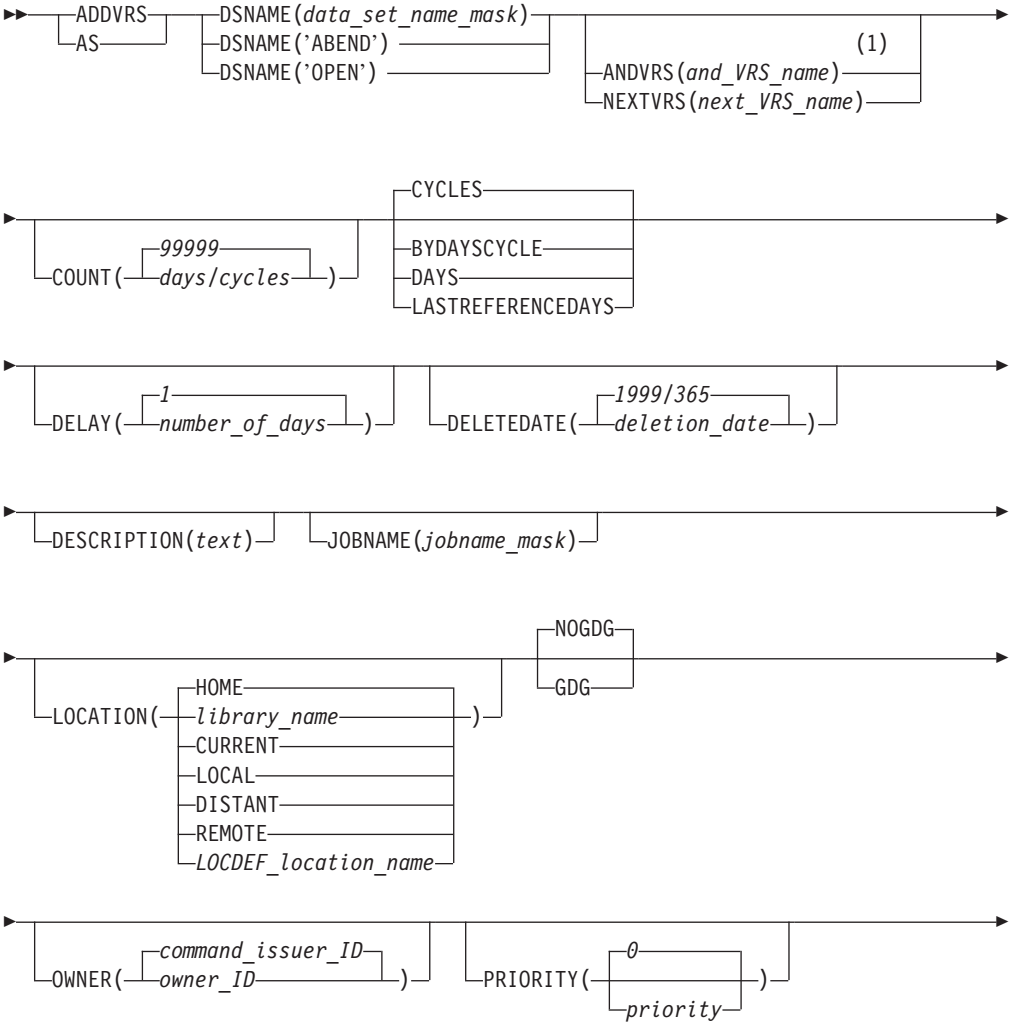
set name in the data set vital record specification. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information on assigning management class names and vital record specification management values.

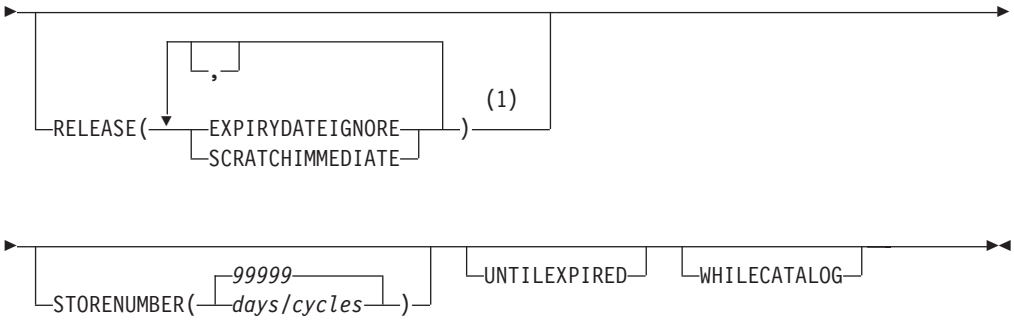
Name Vital Record Specifications
Use the ADDVRS subcommand with the NAME operand to define retention and movement policies by linking name vital record specifications to data set or volume vital record specifications. Although name vital record specifications can define both retention and movement policies, 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.

Syntax

ADDVRS DATASET Syntax



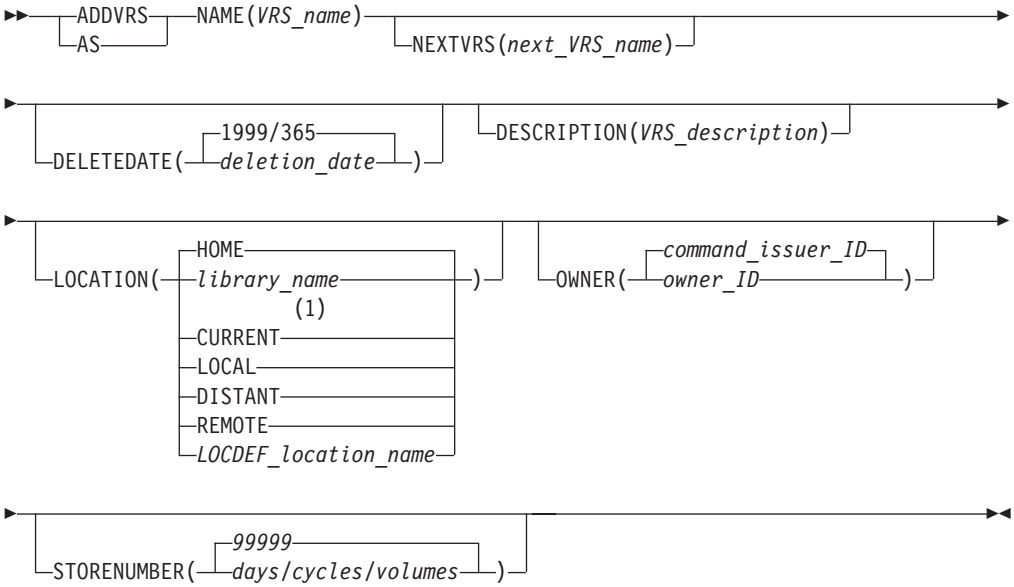


Notes:

1 This operand is honored when you specify the DFSMSrmm parmlib OPTION VRSEL(NEW) operand.

Syntax

ADDVRS Location NAME Syntax

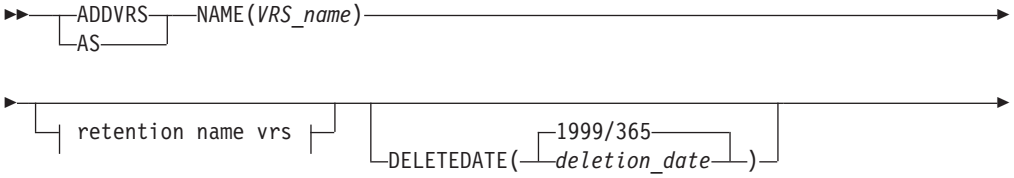


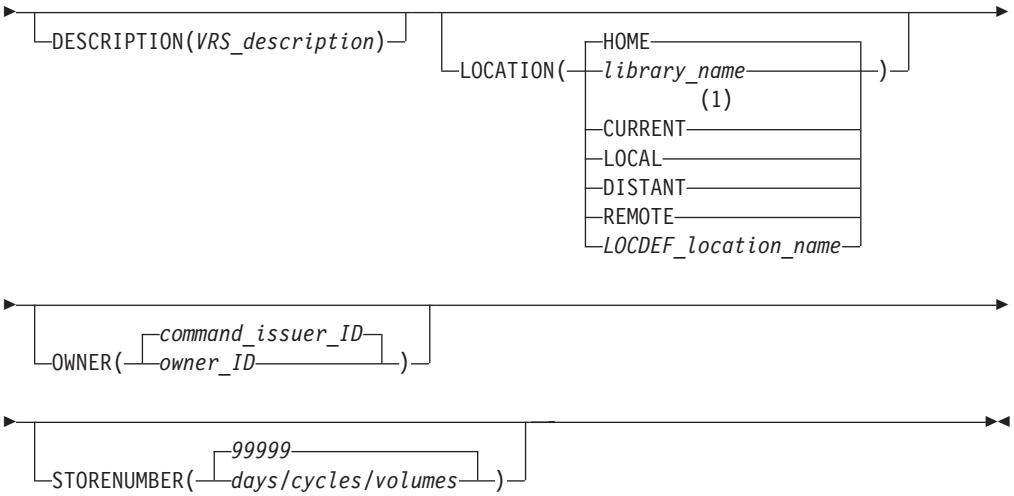
Notes:

1 This operand is honored when you specify the DFSMSrmm parmlib OPTION VRSEL(NEW) operand.

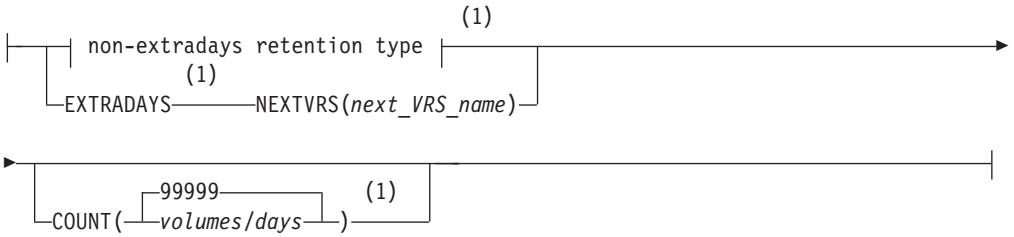
Syntax

ADDVRS Retention NAME Syntax

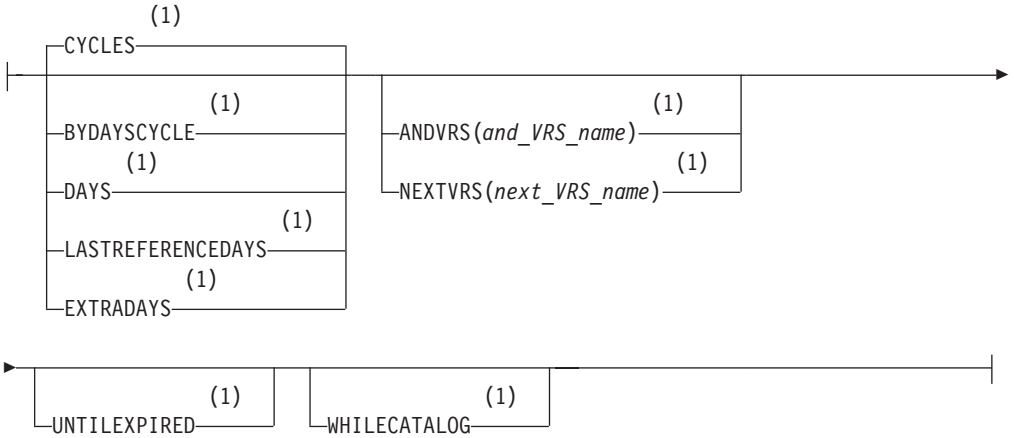




retention name vrs:



non-extradays retention type:

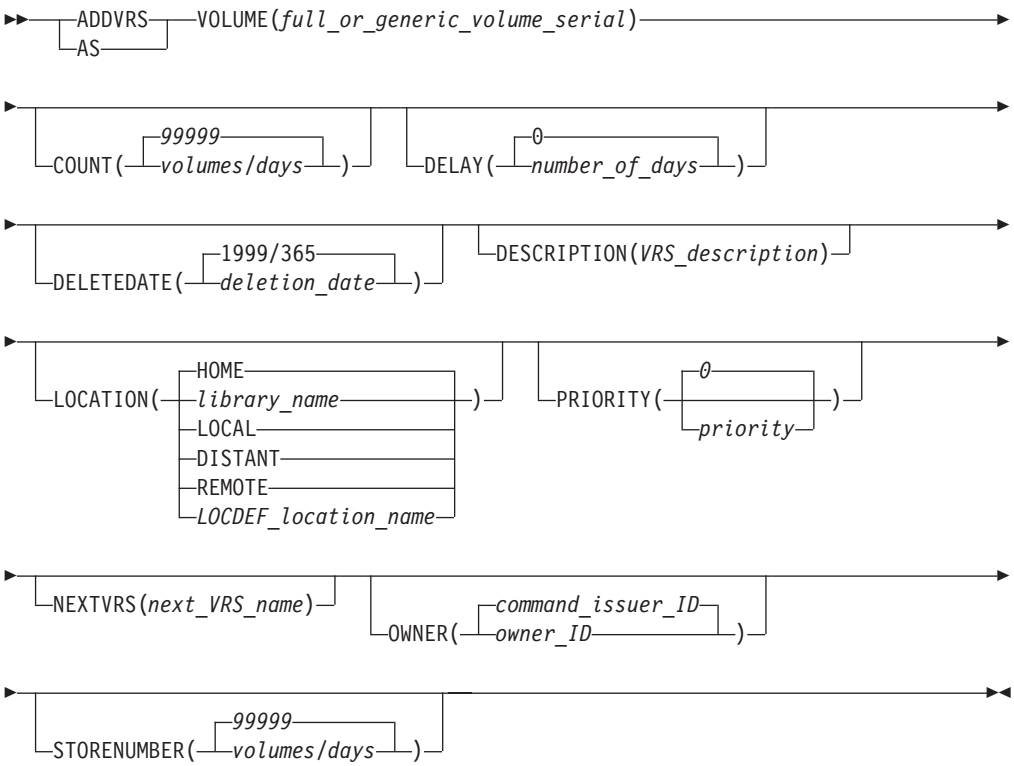


Notes:

1 This operand is honored when you specify the DFSMSrmm parmlib OPTION VRSEL(NEW) operand. When you run inventory management VRSEL(OLD), DFSMSrmm issues return code 4.

Syntax

ADDVRS VOLUME Syntax



Operands

ANDVRS(*and_VRS_name***)**
Specifies creating 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
Specifies retaining 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.

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

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. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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 quotation marks if it contains any special characters or blanks. The default is blanks.

DSNAME(*data_set_name_mask***)**
Specifies 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

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 or as a result of an abend in OPEN, CLOSE, or End of Volume processing. 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 48 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.

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 G*nnnn*V*nn*. 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 *z/OS 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 186 and “CHANGEVOLUME: Changing Volume Information” on page 234 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. You cannot specify a distributed library name.

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***)**
Specifies 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.

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 RELEASE options for data set vital record specifications. For multi-file volumes, DFSMSrmm sets the RELEASE options for the volume if any of the multiple data sets is retained by a vital record specification that contains the release option. Each time vital record processing is performed, DFSMSrmm sets the release options for only those data sets that are still retained by vital record specifications. When a volume is no longer retained by a vital record specification, DFSMSrmm uses the RELEASE options that were set the last time vital record processing was run. Specify the DFSMSrmm EDGRMMxx parmlib OPTION command VRSEL(NEW) operand for DFSMSrmm to process the release option.

EXPIRYDATEIGNORE
Specifies that DFSMSrmm expiration processing should ignore a volume's expiration date when the volume is no longer retained by a vital record specification.

SCRATCHIMMEDIATE
Specifies that the volume can be marked for return to scratch when the only pending action is return to scratch.

When multiple data sets are created on a volume and retained by vital record specifications, DFSMSrmm sets the volume release options if any matching vital record specifications specify that option. DFSMSrmm updates release options each time vital record processing is run.

To remove the EXPIRYDATEIGNORE or SCRATCHIMMEDIATE release options for a volume, remove the RELEASE operand on all the vital record specifications that retain data sets on the volume. Then rerun vital record processing.

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 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 61 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 that is 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, national, or special characters. Enclose it in single quotation marks if it contains any special characters. A generic volume serial number is one-to-five alphanumeric, national, 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 that matches a vital record specification 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 data sets 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. DFSMSrmm retains the data set for the catalog retention period if the data set has never been cataloged. DFSMSrmm does not retain the data set if DFSMSrmm detected that the data set was cataloged and then uncataloged during the catalog retention period.

As a default, DFSMSrmm does not consider the catalog status of a data set when determining the retention.

See “Merging Retention Policies” on page 61 for information about DFSMSrmm processing when UNTILEXPIRED and other retention types like WHILECATALOG are used at the same time.

Examples

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.%%%%TAPE.DATASET') COUNT(99999)-
    LOCATION(LIB1)
```

Task: Add a volume vital record specification. The vital record specification owner's ID is OWNER77 and the volume serial number is 8E1U02. Indicate that the volume is to remain in its home location indefinitely.

Command:

```
RMM ADDVRS VOLUME(8E1U02) OWNER(OWNER77) LOCATION(HOME) COUNT(99999)
```

Task: Define two vital record specifications to move DFSMShsm ABARS 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. Create pseudo-GDGs to retain volumes 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. STORENUMBER and COUNT are default values.

Command

```
RMM ADDVRS DSNAME('HLQ.**') DAYS LOCATION(DISTANT) -
    STORENUMBER(99999) COUNT(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) COUNT(99999) UNTILEXPIRED
```

Task: Retain data sets closed as a result of abend processing by specifying multiple ABEND or OPEN data set vital record specifications. For example, you can define the following ADDVRS subcommands to define vital record specifications for data sets closed by abend processing.

Command

```
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 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

CHANGEDATASET: Changing Data Set Information

Before you begin:

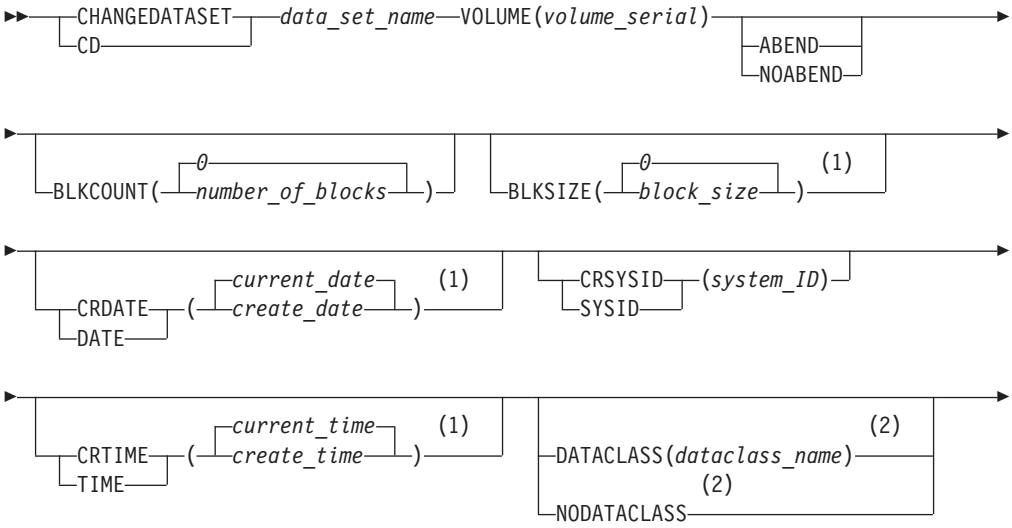
- To use the RMM CHANGEDATASET subcommand, you need UPDATE access to the STGADMIN.EDG.MASTER resource profile to change information in non-restricted fields for your own volumes.
- To use the RMM CHANGEDATASET FORCE subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile and UPDATE access to the STGADMIN.EDG.FORCE resource profile to add a data set to a volume where information was recorded during O/C/EOV processing.
- If the DFSMSrmm parmlib OPTION COMMANDAUTH(DSN) command is in effect, you need UPDATE access to the data set name in the DATASET class.
- You need CONTROL access to the STGADMIN.EDG.MASTER resource profile to change information for data sets other than your own.

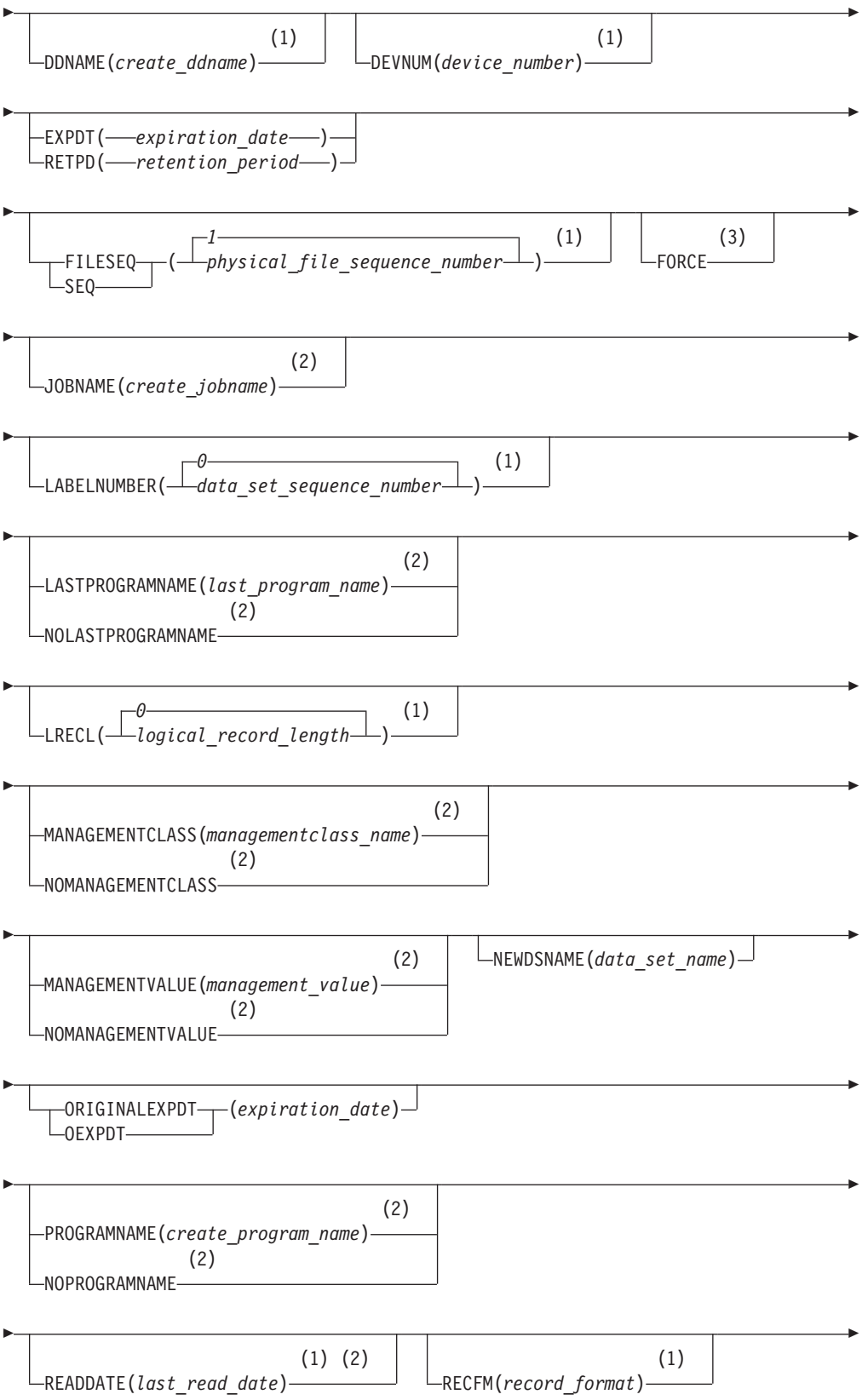
Related Reading: See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

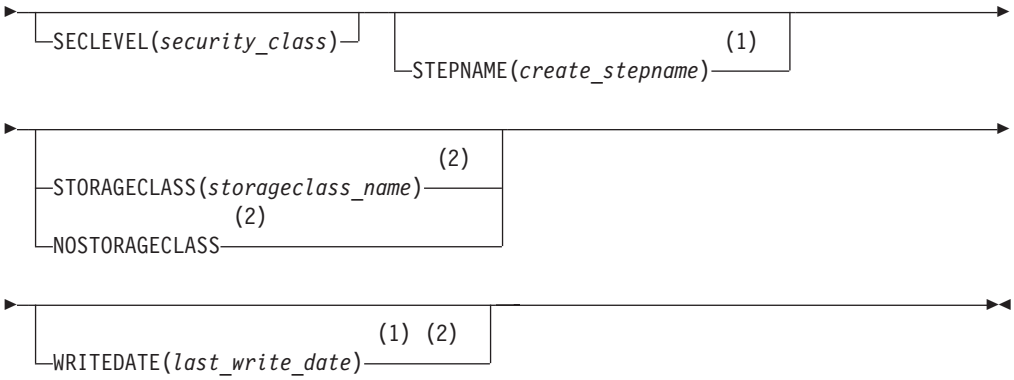
Use the CHANGEDATASET subcommand to update information about data sets defined to DFSMSrmm. Give a data set name and the serial number of the volume where it resides and, optionally, the data set's relative position on the volume. DFSMSrmm assumes that the data set is the first one on the specified volume 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.

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 295 for more information.

Syntax




DDNAME(create_ddname) (1)
DEVNUM(device_number) (1)
EXPDT(expiration_date)
RETPD(retention_period)
FILESEQ
SEQ
(1)
physical_file_sequence_number
FORCE (3)
JOBNAME(create_jobname) (2)
LABELNUMBER(data_set_sequence_number) (1)
LASTPROGRAMNAME(last_program_name) (2)
NOLASTPROGRAMNAME (2)
LRECL(logical_record_length) (1)
MANAGEMENTCLASS(managementclass_name) (2)
NOMANAGEMENTCLASS (2)
MANAGEMENTVALUE(management_value) (2)
NOMANAGEMENTVALUE (2)
NEWDSNAME(data_set_name)
ORIGINALEXPDT
OEXPDT
(expiration_date)
PROGRAMNAME(create_program_name) (2)
NOPROGRAMNAME (2)
READDATE(last_read_date) (1) (2)
RECFM(record_format) (1)



Notes:

1 This operand cannot be specified if DFSMSrmm has automatically recorded information about the volume when a data set on the volume was first opened.

2 This operand can only be specified by a user with CONTROL access to the STGADMIN.EDG.MASTER security resource. Owner authorization does not apply.

3 Specify this operand 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 the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource to use the FORCE operand.

Operands

ABEND
Specifies a data set for special processing. DFSMSrmm marks data sets that were created during a job that abnormally ends so that these data sets can be managed using special policies. Using a vital record specification with the ABEND data set name mask, you can define a policy for these data sets. Use the NOABEND operand if you want to unmark the data set and make it eligible for normal processing.

BLKCOUNT(*number_of_blocks***)**
Specifies 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***)**
Specifies the data set's block size. The minimum allowable decimal value is 0; the maximum allowable decimal value is 999999. DFSMSrmm uses BLKSIZE together with BLKCOUNT to calculate the data set's approximate size when the data set is closed. You can specify this operand if you are authorized to use the FORCE operand. The default value is 0.

CRDATE|DATE(*create_date***|***current_date***)**
Specifies the date when the data set was written to tape.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of

the year, such as 2004/001. The slash is required. You can specify a date in the range between 0000/000 to 9799/365.

- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

You can specify a date in the range between 00000 to 99366.

You can specify this operand if you are authorized to use the FORCE operand.

CRSYSID|SYSID(*creating_system_ID***)**
Specifies the ID of the system on which the data set was created. Specify a one to eight character unique system name.

When you run DFSMSrmm with unshared catalogs, DFSMSrmm uses the CRSYSID of the first file on a volume to determine which the system where the volume should return to scratch.

CRTIME|TIME(*create_time***|***current_time***)**
Specifies the time the data set was written to tape, in the format *hhmmss* where:
- *hh* is hours
- *mm* is minutes
- *ss* is seconds

You can specify a time in the range between 000000 to 235959.

For example, nine o'clock in the morning is 090000. You can specify this operand if you are authorized to use the FORCE operand.

DATACLASS(*dataclass_name***)**
Specifies the data class name. *dataclass_name* is one-to-eight alphanumeric characters and must begin with an alphabetic character. If you change the DATACLASS for the first file on a volume that resides in a VTS, the volume policy is changed too. A DATACLASS can be set for other data sets without any effect to the volume policy.

There is not default value.

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 DD name 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. You can specify this operand if you are authorized to use the FORCE operand.

DEVNUM(*device_number***)**
Specifies 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. You can specify this operand if you are authorized to use the FORCE operand.

EXPDT(*expiration_date***)**
Specifies the date when the data set should be considered for release. The expiration date cannot exceed the maximum retention period (MAXRETPD) specified for your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

Supply the year and day in one of two forms. We recommend that you use the *yyyy/ddd* format rather than the *yyddd* format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

EXPDT is mutually exclusive with RETPD.

FILESEQ(1|*physical_file_sequence_number***)**
Specifies the physical file sequence number. You cannot change the file sequence number. Use this operand to identify the relative position of the data set on the volume. The minimum allowable decimal value is 1. The maximum allowable decimal value is 65535.

The default value is 1.

FORCE
Specifies 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.

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 65535. You can specify this operand if you are authorized to use the FORCE operand. 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.

LASTPROGRAMNAME(*last_program_name***)**
Specifies the name of the program that last referenced the data set.

DFSMSrmm originally records the name of the job step program running at the time the data set is opened for output. A program name is one-to-eight alphanumeric characters, $, #, or @ and must start with an alphabetic character, $, #, or @. You can specify this operand if you have CONTROL access to STGADMIN.EDG.MASTER security resource.

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 99999. You can specify this operand if you are authorized to use the FORCE operand.

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. You can specify management class to define policies for both sms-managed volumes and non-sms-managed volumes. If you change the MANAGEMENTCLASS value for the first file on a volume that resides in a VTS, DFSMSrmm communicates the policy change to the VTS.

There is no default value.

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***)**
Specifies the name of a data set. Use this operand to change the name of a data set in the DFSMSrmm control data set. 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
Specifies the NOABEND condition. Use the NOABEND operand if you want to unmark a data set that DFSMSrmm marks for special processing when a job abnormally ends Using the NOABEND operand makes the data set eligible for normal processing.

NODATASETCLASS
Specifies that you want the DATASETCLASS value set to blanks.

NOLASTPROGRAMNAME
Specifies the NOLASTPROGRAMNAME condition. Specify this operand to remove the name of the program that last referenced the data set.

NOMANAGEMENTCLASS
Specifies that you want the MANAGEMENTCLASS value set to blanks.

NOPROGRAMNAME
Specifies the NOPROGRAMNAME condition. Specify to remove the name of the program that created the data set.

NOSTORAGECLASS
Specifies that you want the STORAGECLASS value set to blanks.

ORIGINALEXPDT/OEXPDT(*expiration_date***)**
Specifies the original JCL expiration date of the data set.

Supply the year and day in one of two forms. We recommend that you use the *yyyy/ddd* format rather than the *yyddd* format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

There is no default.

PROGRAMNAME(*create_program_name***)**
Specifies the name of the program that created the data set. DFSMSrmm records the name of the job step program running at the time the data set is opened for output. A program name is one-to-eight alphanumeric characters, $, #, or @ and must start with an alphabetic character, $, #, or @. You can specify this operand if you have CONTROL access to STGADMIN.EDG.MASTER security resource.

READDATE(*last_read_date***)**
Specifies when the data set was last read.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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. You can specify this operand if you are authorized to use the FORCE operand.

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. You can specify this operand if you are authorized to use the FORCE operand.

RETPD(*retention_period***)**
Specifies the number of days that DFSMSrmm retains the data set before considering it for release. *retention_period* is a one-to-four character decimal number. The retention period is added to today's date to create the expiration date. The expiration date cannot exceed the maximum retention period (MAXRETPD) specified for your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

RETPD is mutually exclusive with EXPDT.

SECLEVEL(*security_class***)**
Specifies 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 287 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. You can specify this operand if you are authorized to use the FORCE operand.

STORAGECLASS(*storageclass_name***)**
Specifies the storage class name. A storage class name is one-to-eight alphanumeric characters and must begin with an alphabetic character. The name you specify must be a single qualifier. Change the STORAGECLASS for the first file on a volume to change the policy for a volume that resides in a VTS.

There is no default value.

SYSID(*SMF_system_ID***)**
Specifies the ID for the system where the data set was created. 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. You can specify SYSID if you are authorized to use the FORCE operand.

VOLUME(*volume_serial***)**
Specifies the serial number of the volume where the data set resides. A volume serial number is one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any 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. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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. You can specify this operand if you are authorized to use the FORCE operand.

Examples

Task: Update the dataclass, managementclass, and storageclass information for a data set named DSBA063.DSN01.

Command:

```
RMM CHANGEDATASET 'DSBA063.DSN01' VOL(BCD001) SEQ(1) -
     STORCLAS(SCATL01) DATACLASS(DCBA063) -
     MANAGEMENTCLASS(MCBA063)
```

Task: Update the dataclass information for a data set named DSBA063.DSN01.

Command:

```
RMM CHANGEDATASET 'DSBA063.DSN01' NODATACLASS
```

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)
```

Task: Change the MANAGEMENTVALUE assigned to a data set and set the creating job name.

Command:

```
RMM CHANGEDATASET 'KEEP.UKRMM.TEAM' VOLUME(OHNO95) -
     MANAGEMENTVALUE(D99002) JOBNAME(WARWKLY)
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

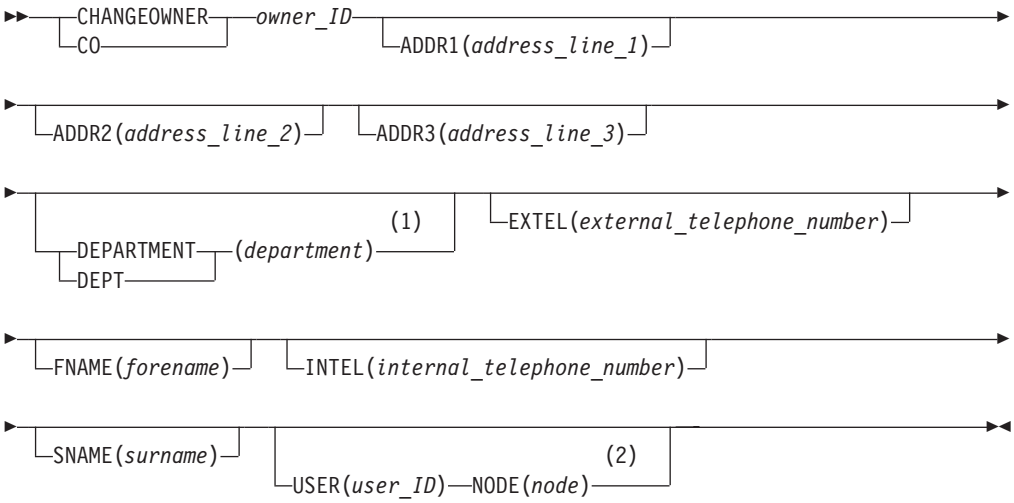
CHANGEOWNER: Changing Owner Information

Before you begin: To use the RMM CHANGEOWNER subcommand:

- You need READ access to the STGADMIN.EDG.MASTER resource profile to change your own owner information.
- You need CONTROL access to the STGADMIN.EDG.MASTER resource profile to update information for any OWNER.

Use the CHANGEOWNER subcommand 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.

Syntax



Notes:

1 The DEPARTMENT operand must contain at least one non-blank character.

2 The values you specify for USER(*user_ID*) and NODE(*node*) work together; if you delete one, you must also delete the other.

Operands

ADDR1(*address_line_1***)**
Specifies the first address line of the owner's address. An address line is one to forty characters enclosed in single quotation marks if it contains any special characters or blanks.

ADDR2(*address_line_2***)**
Specifies the second address line of the owner's address. An address line is one to forty characters enclosed in single quotation marks if it contains any special characters or blanks.

ADDR3(*address_line_3***)**
Specifies the third address line of the owner's address. An address line is one to forty characters enclosed in single quotation marks if it contains any special characters or blanks.

DEPARTMENT/DEPT(*department***)**
Specifies 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 quotation marks if it contains any special characters or blanks.

EXTEL(*external_telephone_number***)**
Specifies the owner's external telephone number. An external telephone number is one to twenty characters enclosed in single quotation marks if it contains any special characters or blanks.

FNAME(*forename***)**
Specifies the owner's forename, or first name. A forename is one to twenty characters enclosed in single quotation marks if it contains any special characters or blanks.

INTEL(*internal_telephone_number***)**
Specifies the owner's internal telephone number. An internal telephone number is one to eight characters enclosed in single quotation marks if it contains any special characters or blanks.

NODE(*node***)**
Specifies 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
Specifies 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***)**
Specifies the owner's surname, or last name. A surname is one to twenty characters enclosed in single quotation marks if it contains any special characters or blanks.

USER(*user_ID***)**
Specifies 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.

Examples

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)
```

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 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

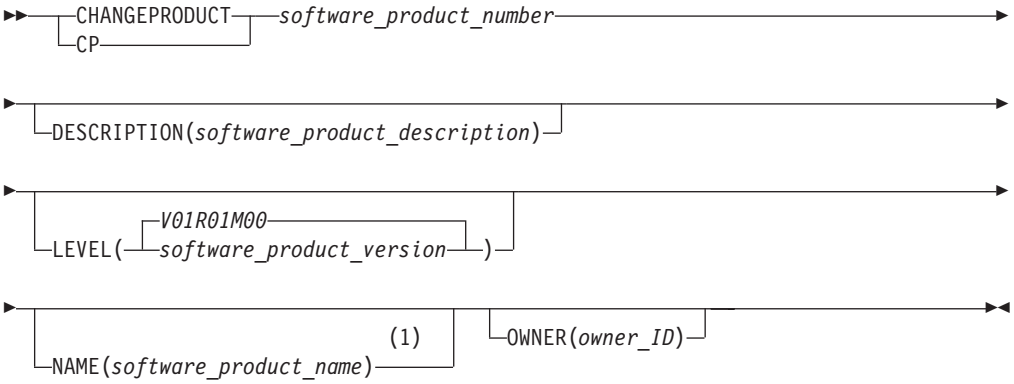
CHANGEPRODUCT: Changing Software Product Information

Before you begin: To use the RMM CHANGEPRODUCT subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile to update a product.

Use the CHANGEPRODUCT subcommand 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 234 and “DELETEVOLUME: Deleting Volume Information” on page 275 for more information.

Syntax



Notes:

1 The NAME operand must contain at least one non-blank character.

Operands

DESCRIPTION(*software_product_description***)**
Specifies the description of the software product. Descriptive text can be one to thirty characters enclosed in single quotation marks if it contains any special characters or blanks.

LEVEL(*software_product_version***)**
Specifies 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.

NAME(*software_product_name***)**
Specifies the software product’s name. A software product name is one to thirty characters enclosed in single quotation marks 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 322 for more information.

OWNER(*owner*)
Specifies 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 quotation marks if it contains any special characters or blanks. This operand is required and must immediately follow the CHANGEPRODUCT subcommand.

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)
```

Return Codes

See Chapter 10, "DFSMSrmm Return Codes and Reason Codes," on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

CHANGEVOLUME: Changing Volume Information

Before you begin:

- To use the CHANGEVOLUME subcommand, you need one of these types of authorization:
  - To update non-restricted volume information that is described in “Changing Non-restricted Volume Information” on page 235:
    - When COMMANDAUTH(OWNER) is in effect; if you own the volume you need UPDATE access to the STGADMIN.EDG.MASTER resource profile; otherwise you need UPDATE access to the STGADMIN.EDG.OWNER.*ownerid* resource profile.
    - When COMMANDAUTH(DSN) is in effect, you need UPDATE access to the STGADMIN.EDG.MASTER resource profile and UPDATE access to the first file data set name in the DATASET class. If there is no first file defined to DFSMSrmm, you need UPDATE access to the volume in the TAPEVOL class.
  - To update restricted fields that is described in “Changing Volume Information Recorded by DFSMSrmm” on page 235 and “Changing Pending Actions” on page 236:
    - You need CONTROL access to the STGADMIN.EDG.MASTER resource profile.
  - To use the CHANGEVOLUME FORCE subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile and UPDATE access to the STGADMIN.EDG.FORCE resource profile.

Related Reading:

- See “Changing Information for Volumes in Multivolume Sets” on page 30 for information about using the DFSMSrmm ISPF dialog to change information for multiple volumes in a multi-volume set.
- See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

Use the CHANGEVOLUME subcommand 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 219 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 27 is information that you can change when you own the volume and you have UPDATE access to STGADMIN.EDG.MASTER or you have UPDATE access to STGADMIN.EDG.OWNER.userid.

Table 27. 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 28 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 28. Volume Information That Can Be Changed Under Special Conditions

ASDATE	ASTIME	COMPACTION	DENSITY
DSNAME	JOBNAME	LABEL	MEDIANAME
NEWVOLUME	ORIGINALEXPDT	READDATE	RECORDINGFORMAT
WRITEDATE			

Changing Volume Information Based on Volume Status: Table 29 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).

Table 29. Volume Information for Volumes in Master and User Status

ACCESS	ADDUSERS	COMPACTION	DELUSERS
DESCRIPTION	DSNAME	EXPDT	FEATCD
JOBNAME	LEVEL	LOANLOC	NUMBER
OWNER	OWNERACCESS	PREVVOL	READDATE
RELEASEACTION	RETPD	SECLEVEL	STORGRP
WRITEDATE			

You can use the operands shown in Table 30 for all volumes regardless of their status:

Table 30. Volume Information for All Volumes

ASDATE	ASTIME	CRDATE	CRSYSID
CRTIME	DENSITY	EJECT	HOME
INIT	LABEL	LOCATION	MEDIATYPE
POOL	RACK	RECORDINGFORMAT	
SPECIALATTRIBUTES	STATUS	USE	

Restriction: 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 31.

Table 31. Volume Information That Can Be Changed with CONTROL Access

ASDATE	ASTIME	AUTOMOVE	BIN
CONFIRMMOVE	CONTAINER	CRDATE	CRSYSID
CRTIME	EJECT	HOME	LOANLOC
LOCATION	MANUALMOVE	MEDIANAME	POOL
RACK	READDATE	STORAGEGROUP	TYPE
WRITEDATE			

Changing Volume Location

To move a volume to any location or from any location without performing vital record processing or storage location management processing, use the DFSMSrmm TSO CHANGEVOLUME subcommand. 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, 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.

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 whereyou 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, 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 79 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 *z/OS DFSMSrmm Implementation and Customization Guide* for information about running DFSMSrmm inventory management.

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 specifyingan asterisk instead of a volume serial number.

You can undo global confirmation for volumes previously confirmed before you run inventory management. 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 287 for more information.

Use CHANGEVOLUME with the NOCONFIRMMOVE or the NOCONFIRMRELEASE operand and a volume serial number of * to undo a global confirmation which changes the status of a move or action from CONFIRMED to PENDING.

Use the READYTOSCRATCH operand toconfirm moves for volumes that are ready to return to scratch.Use the NOTREADYTOSCRATCH operand toconfirm moves for volumes that are private volumes or for volumes with release actions other than 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 73.

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 DFSMSrmm recorded the volume serial number during O/C/EOV processing, use the FORCE operand to change the volume serial number.

Identifying WORM tapes ready for destruction

Because WORM tapes cannot be reused, you cannot return the volumes to the scratch pool when the data has expired. Normally you would expect the volumes to be destroyed once all the data has expired. DFSMSrmm provides support for automating the management of this process using the existing volume release actions. By default DFSMSrmm sets the return to owner release action, but you can alternatively use the replace release action by changing the release action at any time during the volumes life.

You can set the release actions for the volumes to trigger one of the following actions.

- List all the WORM tapes that are pending return to their owner. Pick the volumes from the list and destroy them. After the volumes are destroyed, use the RMM CHANGEVOLUME * CRLSE(RETURN/REPLACE) or RMM CHANGEVOLUME *volser* CRLSE(RETURN/RELACE) command to confirm the release action which deletes the volume information from the DFSMSrmm control data set.
- List all the WORM tapes that are pending replacement. Pick the volumes from the list and destroy them. Create new WORM tapes that use the same set of volume serial numbers. After the volumes are destroyed, use the RMM CHANGEVOLUME * CRLSE(RETURN/REPLACE) or RMM CHANGEVOLUME *volser* CRLSE(RETURN/RELACE) command to confirm the release action which deletes the volume information from the DFSMSrmm control data set.

Changing the status of WORM tapes

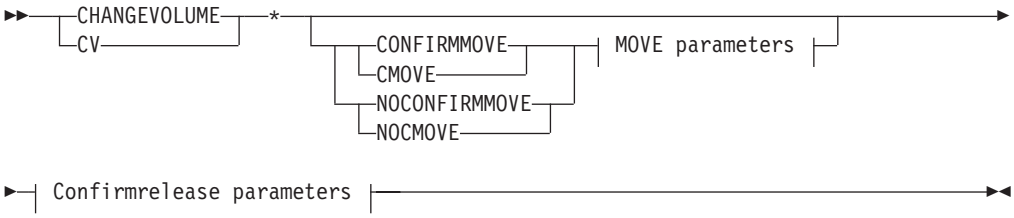
DFSMSrmm prevents WORM tapes from returning to scratch to prevent their misuse. If you inadvertently define a WORM tape as MASTER volume or a USER volume, you cannot scratch the volume once it has been used for input or output. You can change the status of a WORM tape to scratch status using one of these techniques.

1. Use the RMM DELETEVOLUME *volser* FORCE command to delete the volume.
2. Use the RMM ADDVOLUME *volser* STATUS(SCRATCH) command to add the volume to DFSMSrmm with the scratch status.

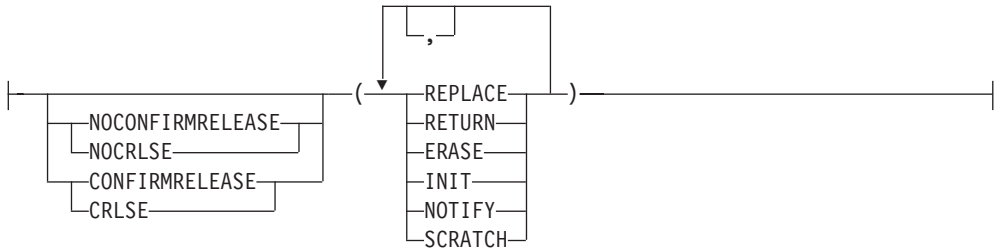
1. Use the RMM CHANGEVOLUME *volser* RELEASE(REPLACE) CONFIRMRELEASE(REPLACE) command to set the release action for the volume to REPLACE and to clear information about the volume from the DFSMSrmm control data set.
2. Optionally, use the RMM CHANGEVOLUME *volser* CONFIRMRELEASE(INIT) command to confirm that the volume is already labeled.
3. Run DFSMSrmm expiration processing to set the volume to scratch status.

Syntax

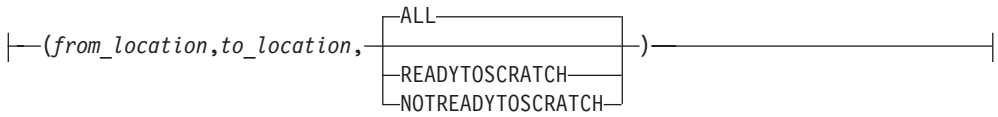
CHANGEVOLUME Global Movement and Confirmrelease Operands



Confirmrelease parameters:

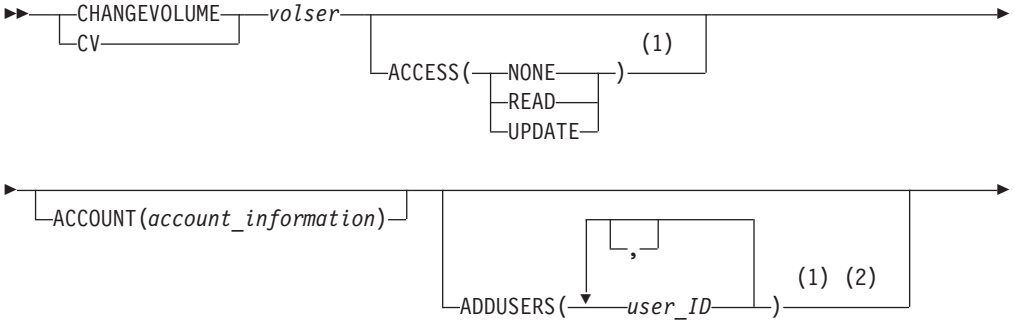


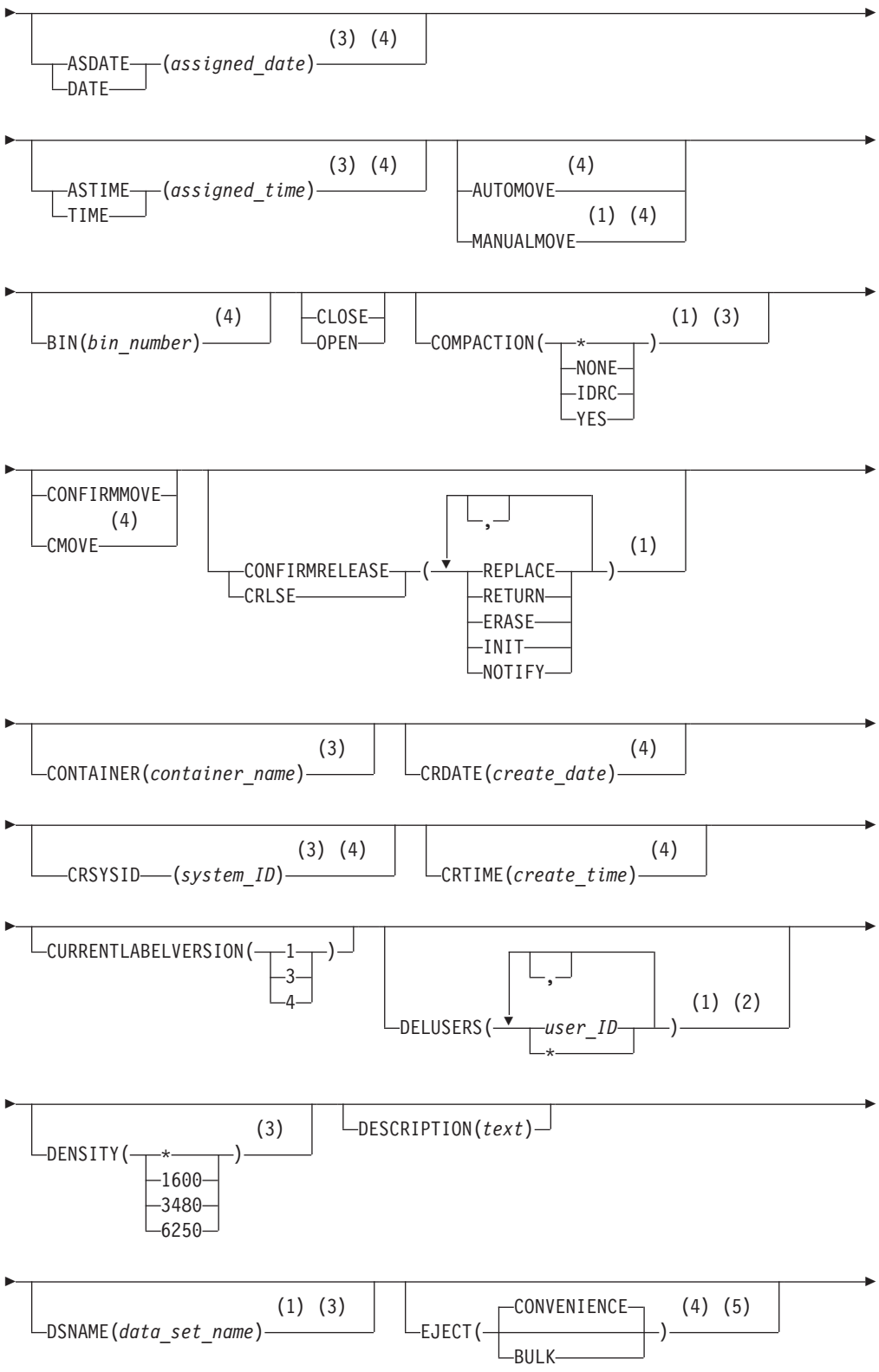
MOVE parameters:

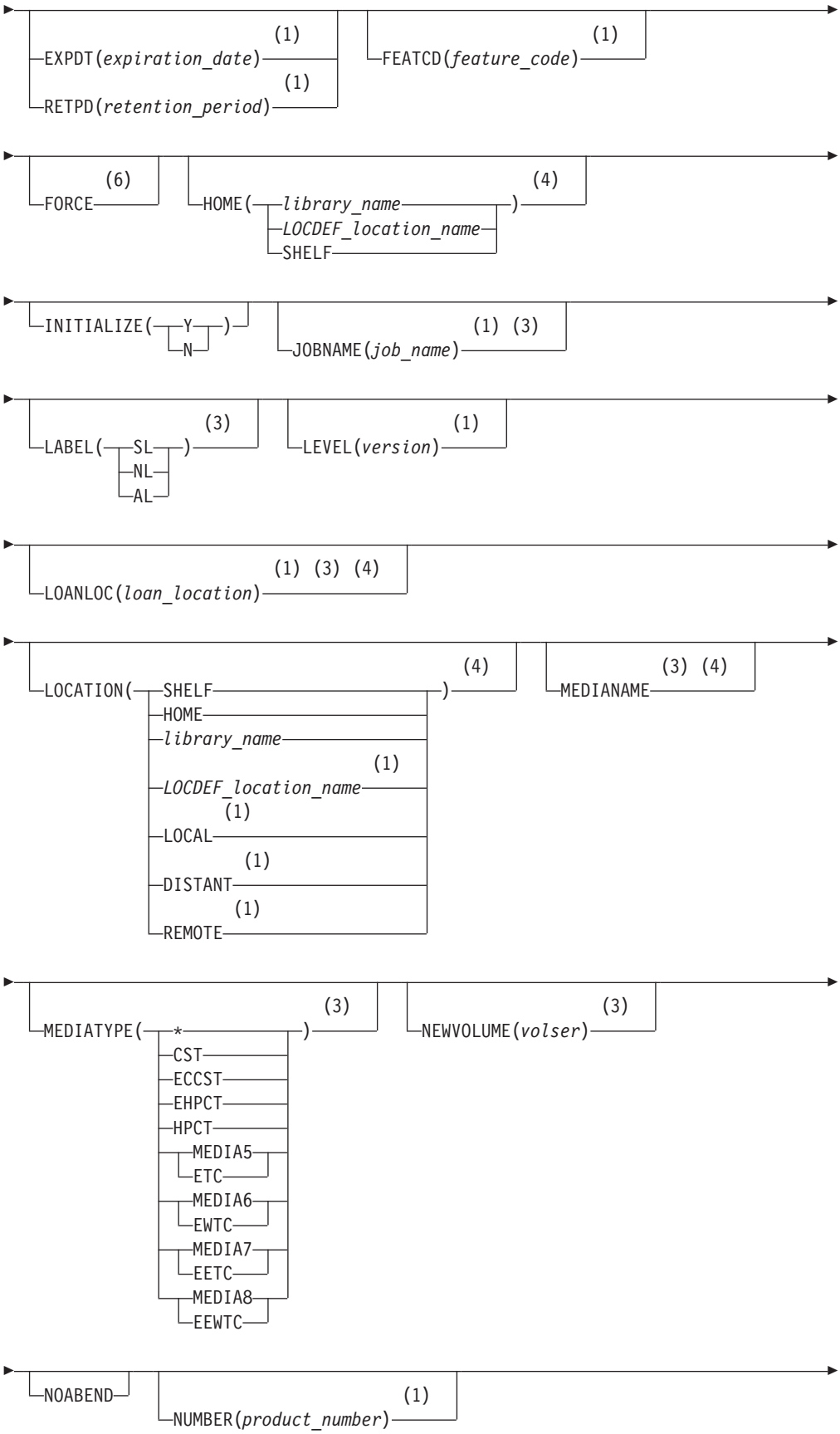


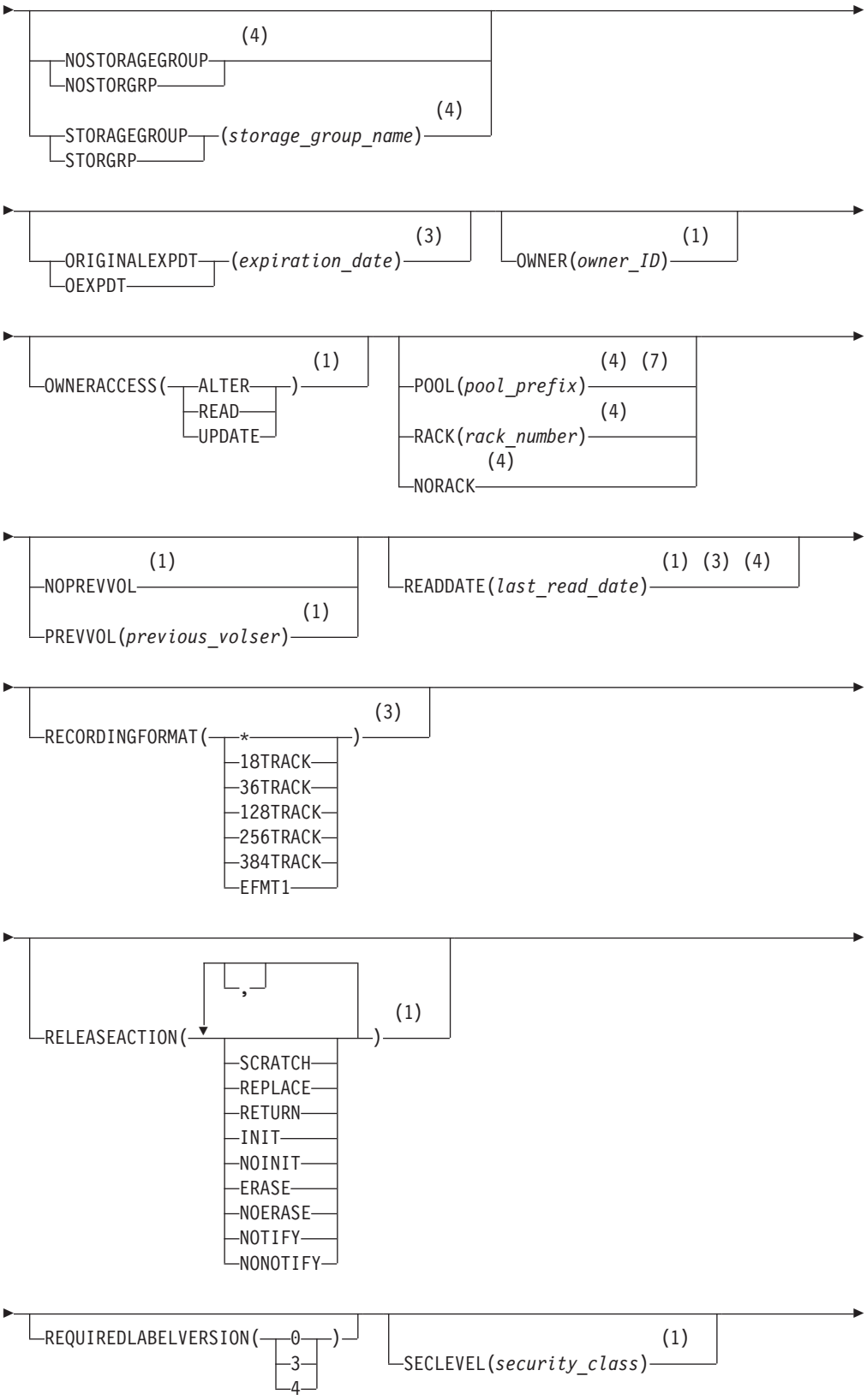
Syntax

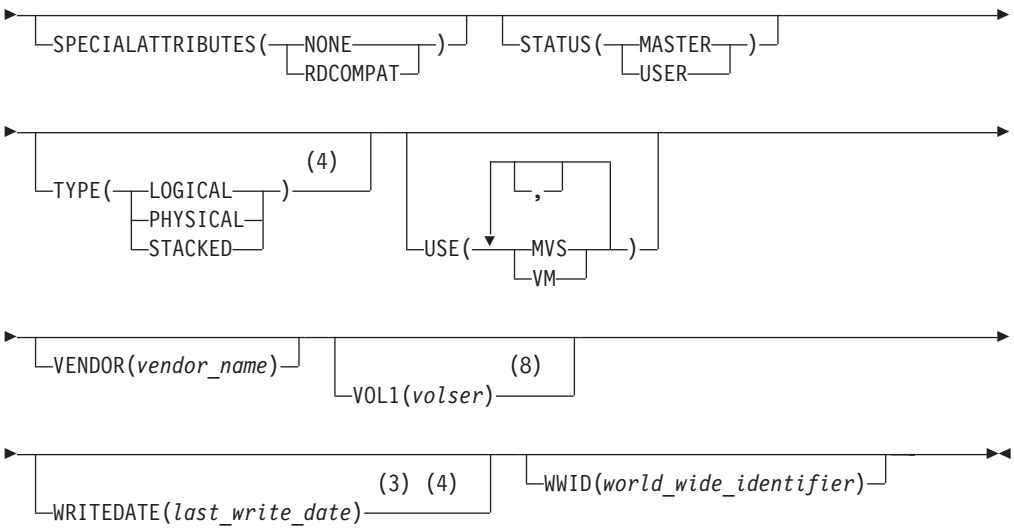
CHANGEVOLUME Specific Volume Operands




ASDATE
DATE
(assigned_date)
(3) (4)
ASTIME
TIME
(assigned_time)
(3) (4)
AUTOMOVE
(4)
MANUALMOVE
(1) (4)
BIN(bin_number)
(4)
CLOSE
OPEN
COMPACTION(
*
NONE
IDRC
YES
)
(1) (3)
CONFIRMMOVE
CMOVE
(4)
CONFIRMRELEASE
CRLSE
(
,
REPLACE
RETURN
ERASE
INIT
NOTIFY
)
(1)
CONTAINER(container_name)
(3)
CRDATE(create_date)
(4)
CRSYSID
(system_ID)
(3) (4)
CRTIME(create_time)
(4)
CURRENTLABELVERSION(
1
3
4
)
DELUSERS(
,
user_ID
*
)
(1) (2)
DENSITY(
*
1600
3480
6250
)
(3)
DESCRIPTION(text)
DSNAME(data_set_name)
(1) (3)
EJECT(
CONVENIENCE
BULK
)
(4) (5)


EXPDT(expiration_date) (1)
RETPD(retention_period) (1)
FEATCD(feature_code) (1)
FORCE (6)
HOME(library_name LOCDEF_location_name SHELF) (4)
INITIALIZE(Y N)
JOBNAME(job_name) (1) (3)
LABEL(SL NL AL) (3)
LEVEL(version) (1)
LOANLOC(loan_location) (1) (3) (4)
LOCATION(SHELF HOME library_name LOCDEF_location_name (1) LOCAL (1) DISTANT (1) REMOTE) (4)
MEDIANAME (3) (4)
MEDIATYPE(* CST ECCST EHPCT HPCT MEDIA5 ETC MEDIA6 EWTC MEDIA7 EETC MEDIA8 EEWTC) (3)
NEWVOLUME(volser) (3)
NOABEND
NUMBER(product_number) (1)


NOSTORAGEGROUP
NOSTORGRP
(4)
STORAGEGROUP
STORGRP
(storage_group_name)
(4)
ORIGINALEXPDT
OEXPDT
(expiration_date)
(3)
OWNER(owner_ID)
(1)
OWNERACCESS(
ALTER
READ
UPDATE
)
(1)
POOL(pool_prefix)
(4) (7)
RACK(rack_number)
(4)
NORACK
(4)
NOPREVVOL
(1)
PREVVOL(previous_volser)
(1)
READDATE(last_read_date)
(1) (3) (4)
RECORDINGFORMAT(
*
18TRACK
36TRACK
128TRACK
256TRACK
384TRACK
EFMT1
)
(3)
RELEASEACTION(
,
SCRATCH
REPLACE
RETURN
INIT
NOINIT
ERASE
NOERASE
NOTIFY
NONOTIFY
)
(1)
REQUIREDLABELVERSION(
0
3
4
)
SECLEVEL(security_class)
(1)



Notes:

1 This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

2 You can specify a maximum of 12 user IDs.

3 You can specify this operand only if you also specify the FORCE operand. To use the FORCE operand, you must have CONTROL access to the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource.

4 This operand can be specified if the user has CONTROL access to STGADMIN.EDG.MASTER resource.

5 This operand can be specified only if the volume resides in a system-managed library.

6 This operand 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 the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource to use the FORCE operand.

7 This operand cannot be specified if the volume resides in an IBM Tape Library Dataserver.

8 This operand cannot be specified for SCRATCH volumes, LOGICAL volumes, STACKED volumes, or NOLABEL volumes.

Operands

ACCESS(NONE|READ|UPDATE)

Specifies 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***)**
Specifies the accounting information recorded for the volume. Accounting information is one to forty characters enclosed in single quotation marks 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.

ADDUSERS(*user_ID,user_ID,...***)**
Specifies 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***)**
Specifies 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. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

You can specify the dates 99365 and 99366 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.

ASTIME|TIME(*assigned_time***)**
Specifies 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
Specifies that you want to return the volume to DFSMSrmm inventory management automatic movement control.

BIN(*bin_number***)**
Specifies the shelf location in a storage location. 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.

CLOSE
Specifies to reset the volume OPEN condition. You can use this operand for any volume. For logical and physical volumes, use this operand to indicate that all data sets on the volume have been closed successfully. For stacked volumes, use this operand to indicate that no more volumes will be exported to it. A stacked volume is normally closed by the completion of export processing. Use this operand when you want to mark a stacked volume as ready to move. DFSMSrmm can move stacked volumes automatically only when they are marked closed.

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(*from_location,to_location***, ALL, READYTOSCRATCH,NOTREADYTOSCRATCH)**
Specifies that a pending move for a volume has taken place.

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.

To confirm volume movement for volumes moving from one location to another location, you can specify a *from_location*, *to_location*. To confirm volume movement for all locations, specify ALL. To confirm volume movement for volumes based on their status, you can specify 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
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
- **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, 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 287 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,SCRATCH)
Specifies that you have performed the requested actions and that release processing can continue. You only need to confirm those actions which you perform manually. DFSMSrmm confirms those actions it performs 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.

SCRATCH You have performed all activities necessary prior to returning the volume to scratch status.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

CONTAINER(*container_name***)**
Specifies the name of the stacked volume containing this volume. The container can be a physical stacked volume or any other receptacle where a volume can reside. The value can be any alphanumeric or special characters up to 6 characters in length. To clear the container value, specify CONTAINER(' '). For VTS support, when a volume is removed from a container and you have enabled stacked volume support, DFSMSrmm sets the volume location based on the HOME location of the stacked volume unless you provide a new location using the LOCATION operand.

When a logical volume is added to a stacked volume container using the DFSMSrmm TSO subcommands, DFSMSrmm allows this only if the volume was once exported to that stacked volume and the stacked volume has not been reused. DFSMSrmm tracks unique information to ensure that you can associate a logical volume with the correct stacked volume container. Reusing the stacked volume container or writing to the logical volume prevents DFSMSrmm from associating the logical volume with the stacked volume container.

When stacked volume support is enabled, you can add container information for a volume if the stacked volume you specify is already defined to DFSMSrmm. If the stacked volume is not defined, DFSMSrmm fails the request.

CRDATE(*create_date*)
Specifies the date when the volume was created.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required. You can specify a date in the range between 0000/000 to 9799/365.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

You can specify a date in the range between 00000 to 99366.

CRSYSID
Specifies the ID of the system on which the volume was created. Specify a unique system name one-to-eight characters long.

CRTIME(*create_time*)

Specifies the time when the volume was created. The CRTIME format is *hhmmss* where:
- *hh* is hours
- *mm* is minutes
- *ss* is seconds

You can specify a time in the range between 000000 to 235959.

For example, nine o'clock in the morning is 090000.

CURRENTLABELVERSION(1|3|4)
Specifies that the ISO/ANSI label version set for the volume should be changed.

DELUSERS(USER_ID LIST|*)
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. If you need to delete all user IDs from the list, use an asterisk instead of the list. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

DENSITY(*|1600|3480|6250)
Specifies 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*)
Specifies the volume description. Descriptive text is one to thirty characters enclosed in single quotation marks if it contains any special characters or blanks.

DSNAME(*data_set_name*)
Specifies 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 a system-managed tape library. Specify this operand on a system with access to the library in which the volume resides. DFSMSrmm marks the volume as intransit.

Use EJECT(CONVENIENCE) to eject the volume to the convenience output station. Use EJECT(BULK) to eject the volume to the high capacity output station. If you use EJECT(BULK) for a volume residing in a manual tape library, 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 processes 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. When you specify EJECT for logical volumes that are scratch volumes, the volume serial number is removed from the inventory in the Library Manager database and from the TCDB.

EXPDT(*expiration_date***)**
Specifies the volume expiration date which is the date when the volume should be considered for release. The value cannot exceed the maximum retention period (MAXRETPD) set by your installation in parmlib member EDGRMM*xx*.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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***)**
Specifies 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
Specifies information that DFSMSrmm recorded during O/C/EOV processing. You can specify the FORCE operand to

1. Change information that DFSMSrmm records during O/C/EOV processing.
2. Mark a volume as ejected from a system-managed tape library even when the library is not known or offline when you combine FORCE with the EJECT operand.
3. Mark a volume as moved from a system-managed tape library when you combine FORCE with the LOCATION operand. Using the FORCE operand does not prevent updates to the TCDB. If the TCDB is available and can be updated, DFSMSrmm updates it. Using the FORCE operand allows the command to complete successfully even if the TCDB is not available or incorrect.

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.

HOME(*library_name***|***LOCDEF_location_name***|SHELF)**
Specifies the location where the volume returns when it is no longer retained by a vital record specification. This location can be either an automated or a manual tape library, or a shelf location in a non-system-managed library. This storage location can be defined to DFSMSrmm as a home location. This value does not cause any volume movement.

LOCDEF_location_name indicates that the volume is stored in a storage location. When you store volumes in a storage location as their home location, you enable volumes to return to scratch in the named storage location. Storing volumes in this way allows you to avoid using the location SHELF.

DFSMSrmm processes the HOME operand before the LOCATION operand.

You specify SHELF or a VTS library name for a logical volume.

A home location name is one-to-eight alphanumeric characters starting with a nonnumeric character.

INITIALIZE(Y|N)
Specifies whether the volume should be initialized. Specify INITIALIZE(Y) for all volumes, except stacked volumes, to indicate that the volume requires initializing. You cannot use the volume until you confirm that the volume is initialized. Specify INITIALIZE(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, DFSMSrmm defers initialization to the DFSMSdfp labeling program. If the volume is later ejected without being initialized, DFSMSrmm reinstates the initialization. If you specify INITIALIZE(Y) for volumes residing in an automated tape library, the action is set pending and you must initialize the volume before it can be used.

Specify INITIALIZE(Y) to relabel a duplicate volume so that DFSMSrmm no longer manages it as a duplicate.

DFSMSrmm accepts the abbreviation INIT.

JOBNAME(*job_name***)**
Specifies 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***)**
Specifies 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***)**
Specifies 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 quotation marks 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. If a volume is stored in a system-managed tape library, DFSMSrmm automatically ejects the volume if you define a loan location. You can clear the loan location by specifying a blank loan location.

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***)**
Specifies 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, DFSMSrmm uses the LOCATION value to set the home location name in the DFSMSrmm control data set if the HOME location is not a storage location.

You can use the LOCATION operand to correct the location information recorded by DFSMSrmm when it does not match the library name in the TCDB. Using the LOCATION operand, you can correct the DFSMSrmm control data set without physically moving the volume.

If the location name you specify does not match the volume's current location, DFSMSrmm attempts to move the volume. If the current location for the volume is a system-managed library, DFSMSrmm attempts to eject the volume. To move the volume, DFSMSrmm sets the required location for the volume. If the volume is a logical volume and stacked volume support is enabled, or a volume that is moving to a shelf-managed storage location, DFSMSrmm inventory management performs any required movement for the volume. In all other cases, DFSMSrmm sets the volume destination and ejects the volume if necessary. Non-scratch logical volumes cannot be ejected.

If you have enabled stacked volume support, DFSMSrmm inventory management uses the required location to determine if the stacked volume

container is moved. You can use the required location for logical volumes to build an export list. If stacked volume support is not enabled, you can use the destination to build an export list.

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 returns from its current location to the location identified by the HOME operand.

library_name
To indicate that the volume be stored in a specific system-managed library.

This library can be either a manual tape library 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, DFSMSrmm adds the volume to the TCDB. If the library is an automated tape library and the volume does not currently reside in the automated tape library, DFSMSrmm sets a volume move in progress to get the volume moved to the automated tape library. You cannot specify a distributed library name.

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
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 moveby specifying the volume's current location.

MANUALMOVE
Specifies to turn off DFSMSrmm automatic move processing or to cancel a pending move.

MEDIANAME
Specifies 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|MEDIA5|MEDIA6|MEDIA7|MEDIA8)
Specifies the volume's physical media type. Use one of the following:

*	The volume is not a cartridge.
CST	Cartridge System Tape
ECCST	Enhanced Capacity Cartridge System Tape
EHPCT	Extended High Performance Cartridge Tape

HPCT High Performance Cartridge Tape

MEDIA5/ETC IBM TotalStorage Enterprise Tape Cartridge

MEDIA6/EWTC
IBM TotalStorage Enterprise WORM Tape Cartridge 3592

MEDIA7/EETC
IBM TotalStorage Enterprise Economy Tape Cartridge 3592

MEDIA8/EEWTC
IBM TotalStorage Enterprise Economy WORM Tape Cartridge 3592

Recommendation: Specify the known media type for system-managed, non-scratch volumes. If you use an incorrect value, the volume can be mounted on a tape drive that can neither read nor write to the volume. If you do not specify a value, DFSMSrmm sets a default value that is valid for the media type you specify.

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.

See Table 6 on page 27 for information about how DFSMSrmm assigns MEDIATYPE and RECORDINGFORMAT.

There is no default value.

NEWVOLUME(*volser***)**
Specifies the volume serial number of a volume in the DFSMSrmm control data set. 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. You cannot use this operand to change the volume serial number for a logical or stacked volume. You can use this operand to identify a duplicate 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.

To change a volume serial number that was recorded by DFSMSrmm during O/C/EOV processing, use the FORCE operand to force the change.

NOABEND
Specifies to reset the ABEND flag for a volume.

NOCONFIRMMOVE/NOCMOVE(*from_location***,***to_location***,ALL)**
Specifies 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)
```

NOCONFIRMRELEASE/NOCRLSE(REPLACE, RETURN, ERASE, INIT, NOTIFY, SCRATCH)

Specifies 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

SCRATCH You have not performed all return to scratch cleanup actions required for the volume

You can only reverse a previous action confirmation as long as inventory management has not started.

NOPREVVOL
Specify to remove a single volume from the end of a multivolume chain.

NORACK
Specifies to remove an assigned rack number from any volume. This allows the volume to be entered as a logical volume.

NOSTORAGEGROUP
Specifies to remove the name of the storage group to which the volume belongs. You can use this operand to assign a null storage group for a system-managed volume, or to remove an existing storage group name for a non-system managed volume.

NUMBER(*product_number***)**
Specifies the number of a software product with which the volume is associated. A software product number is one to eight characters enclosed in single quotation marks if it contains any special characters or blanks.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

OPEN
Specifies to set the volume OPEN condition. You can specify this operand for any type of volume. For a logical or physical volume, you are specifying that a data set is still open or has been left open by an application. Use this operand when you want to mark a stacked volume as not ready to move. A stacked volume is normally opened by the start of export processing and closed by the completion of export processing. DFSMSrmm can move stacked volumes automatically only when they are marked closed.

ORIGINALEXPDT/OEXPDT(*expiration_date***)**
Specifies the original JCL expiration date of the volume. It should be the highest original expiration date of all the files on the volume.

Supply the year and day in one of two forms. We recommend that you use the *yyyy/ddd* format rather than the *yyddd* format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

There is no default.

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.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

OWNERACCESS(ALTER|READ|UPDATE)
Specifies 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).

POOL(*pool_ID***)**
Specifies the pool ID of the pool where the volume is stored in the removable media library. A pool ID is one-to-five alphanumeric, national, or special characters followed by an asterisk. The pool ID must be defined by your installation. Enclose it in quotation marks if it contains any special characters. If you supply a pool ID, do not give a rack number.

You cannot use this operand for a volume residing in an automated tape library or a manual tape library. POOL cannot be used with RACK.

PREVVOL(*previous_volser***)**
Specifies 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, national, or special characters. Enclose it in quotation marks if it contains any special characters.

To remove a single volume from the end of a multivolume chain, specify PREVVOL('') or use the NOPREVVOL operand as shown in the example on page 262.

This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

RACK(*rack_number***)**
Specifies the shelf location of a volume in the removable media library. A full rack number is one-to-six alphanumeric, national, or special characters. Enclose it in quotation marks if it contains any special characters. The rack number that 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 cannot be used with POOL.

READDATE(*last_read_date***)**
Specifies when the volume was last read.

Supply the year and day in one of two forms.

Recommendation: Use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.

– DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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| 256TRACK|384TRACK|EFMT1)

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.

18TRACK
Data has been written to the volume in 18 track format. Recording format 18TRACK is valid with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

36TRACK
Data has been written to the volume in 36 track format. Recording format 36TRACK is valid with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

128TRACK
Data has been written to the volume in 128 track format. Recording format 128TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

256TRACK
Data has been written to the volume in 256 track format. Recording format 256TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

384TRACK
Data has been written to the volume in 384-track format. A recording format of 384TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

EFMT1
Data has been written to the volume in Enterprise Format 1 recording technology. You can only specify EFMT1 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), and MEDIATYPE(MEDIA8).

You must specify the RECORDINGFORMAT operand when you are changing information about volumes that reside in a manual tape library.

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.

See Table 6 on page 27 for information about how DFSMSrmm assigns MEDIATYPE and RECORDINGFORMAT.

There is no default value.

RELEASEACTION(SCRATCH,REPLACE,RETURN,INIT,NOINIT,ERASE, NOERASE,NOTIFY,NONOTIFY)

Specifies 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.

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)
```

REQUIREDLABELVERSION(0|3|4)
Specifies 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.

RETPD(*retention_period***)**
Specifies 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. The value cannot exceed the maximum retention period (MAXRETPD) set by your installation in parmlib member EDGRMM*xx*. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

RETPD is mutually exclusive with EXPDT. This operand is ignored for scratch volumes unless you use either STATUS(MASTER) or STATUS(USER).

SECLEVEL(*security_class***)**
Specifies 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 287 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.

STATUS(MASTER|USER)
Specifies 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.

USER To indicate that the volume can be overwritten by any data set.

STORAGEGROUP(*storage_group_name***)**
Specifies the SMS-defined storage group to which the volume belongs.

A storage group name is one-to-eight alphanumeric characters. A storage group name can be a value that matches to a VLPOOL NAME value but does not need to be defined on a VLPOOL definition.

For volumes in a system-managed library, DFSMSrmm uses the current location or the location specified on the command to validate the specified storage group. For volumes with a system-managed home location defined, DFSMSrmm uses the home location for validation. For other volumes, any value you specify is accepted as long as the value is valid in the current SMS configuration.

A storage group name can be assigned to any volume, even a scratch volume. The storage group name can be used for scratch pooling, except when the volume is in a system-managed automated tape library. For system-managed scratch volumes, the storage group name is not maintained in the TCDB because it is not supported by SMS tape processing.

DFSMSrmm accepts the abbreviation STORGRP.

TYPE(LOGICAL|PHYSICAL|STACKED)
Specify to change the type that is defined to DFSMSrmm.

TYPE(LOGICAL) and TYPE(STACKED) are not allowed if the LOCATION specifies a system-managed library that is not a VTS.

TYPE(LOGICAL) is not allowed if the volume has a rack number which is different from the volume serial number. If the rack number matches the volser, the rack number is removed from the volume and left empty.

TYPE(PHYSICAL) is not allowed if the LOCATION specifies a system-managed library that is a VTS.

USE(MVS,VM)
Specifies the operating systems where the volume can be used. The values can be MVS and VM. You can select both by specifying MVS,VM with a comma as a separator.

VENDOR(*vendor_name***)**
Specifies the manufacturer or supplier of the volume. The vendor value is 1 to 8 alphanumeric or special characters that you can set or change at any time. DFSMSrmm never changes or uses this value. You can use it for reporting purposes and tracking supplier information for batches of volumes.

There is no default.

*volser,**
Specifies the serial number of the volume being changed. A volume serial number is one-to-six alphanumeric, national, or special characters. Enclose it in quotation marks if it contains any special characters. When you are changing information for volumes that reside in a manual tape library or adding system-managed volumes in an automated tape library, you can only specify volume serial numbers that consist of alphanumeric 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.

When you want to change a volser that is defined with volser *, you must specify CHANGEVOLUME '* ' to differentiate the command from the global CONFIRMRELEASE or CONFIRMMOVE command. At least one space (blank) is required after the asterisk that is enclosed in quotation marks.

VOL1(*volser***)**
Specifies a standard label volume serial number. Use this operand to change information that DFSMSrmm recorded. It does not change the tape label information. The variable *volser* is one-to-six alphanumeric, national, or special characters. You cannot use the VOL1 operand for scratch volumes, logical volumes, stacked volumes, or NL volumes. To clear the VOL1 volser for a volume, specify the volume serial number as the VOL1 volser value.

There is no default.

WRITEDATE(*last_write_date***)**
Specifies when the volume was last written to.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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.

WWID(*world_wide_identifier***)**
Specifies the unique world wide identifier set on the volume by the

manufacturer. The world wide ID is 12 hexadecimal characters. You can specify 24 characters 0-9 and A-F. You can specify the WWID only if it is not already set. If the WWID is already set, you cannot change it.

In some publications, the world-wide unique cartridge identifier (WWCID) may also be referred to as the world-wide identifier (WWID), as in this document.

There is no default.

Examples

Task: To clear the VOL1 label that is recorded by DFSMSrmm for the volume ABC001, issue the RMM CHANGEVOLUME subcommand with a VOL1 value that matches the volume.

Command:

```
RMM CHANGEVOLUME ABC001 VOL1(ABC001)
```

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)
```

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 retains volume 8E1U02 for another five days. This example shows how you can remove the pending release status from the volume so that the volume is not returned to scratch status during the next inventory management run.

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 named MYATL.

Command:

```
RMM CHANGEVOLUME ABC123 LOC(MYATL)
```

Then enter the volume into 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, 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('')
```

Command:

```
RMM CHANGEVOLUME ABC123 NOPREVVOL
```

Return Codes

See Chapter 10, "DFSMSrmm Return Codes and Reason Codes," on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEBIN: Deleting Bin Number Information

Before you begin: To use the RMM DELETEBIN subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

The DELETEBIN subcommand is an alias for the DELETERACK subcommand. See “DELETERACK: Deleting Shelf Location Information” on page 272 for the combined description of the DELETERACK and DELETEBIN operands.

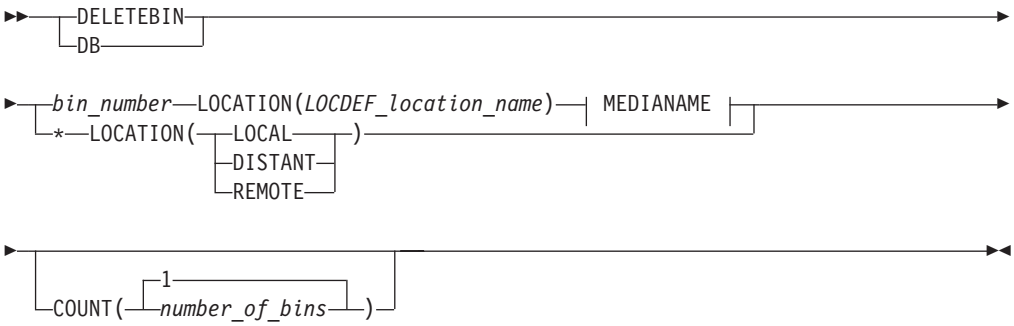
DFSMSrmm defines shelf space in storage locations as bin numbers. Use the DELETEBIN subcommand to delete information about shelf locations you are no longer using in built-in or installation defined storage locations.

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 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.

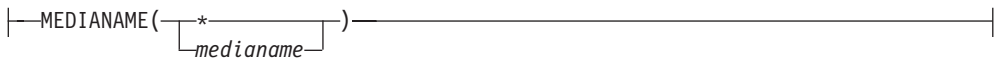
To delete one or more bin numbers from an installation-defined storage location, provide the initial bin number, the location name, and the media name. Specify a COUNT value to indicate how many bin numbers DFSMSrmm deletes. DFSMSrmm deletes bin numbers starting from the initial bin number and deletes 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 DELETEBIN 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.

Syntax



MEDIANAME:



Operands

*bin_number|**

Specifies the shelf location in a storage location. Immediately following the DELETEBIN 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***|*)**

Specifies 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.

Examples

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 DELETEBIN A00001 LOCATION(MYLOC) COUNT(8) MEDIANAME(SQUARE)
```

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 DELETEBIN * 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 DELETEBIN 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 DELETEBIN * LOCATION(REMOTE) COUNT(nnnnn)
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin:

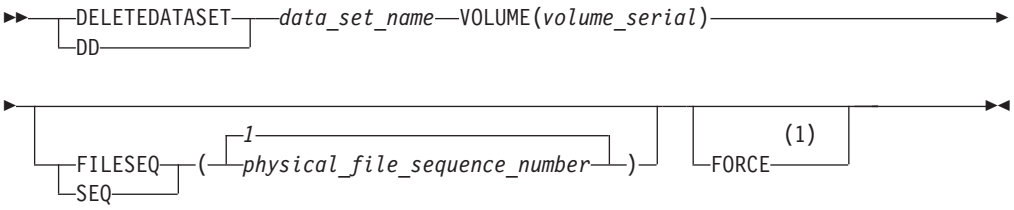
- To use the RMM DELETEDATASET subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.
- To use the RMM DELETEDATASET FORCE subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile and UPDATE access to the STGADMIN.EDG.FORCE resource profile.

Use the DELETEDATASET subcommand to delete data set information recorded by DFSMSrmm. DFSMSrmm also uncatalogs the data set and any other data sets recorded on the same volume that have 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 that 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.

Restriction: You cannot delete any data set on a volume when DFSMSrmmhas recorded information about the data set during O/C/EOV processing.

Syntax



Notes:

1 Specify the FORCE operand to delete a data set for a volume where information was recorded by DFSMSrmm during O/C/EOV processing. You must have CONTROL access to the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource to use the FORCE operand.

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 65535.

The default value is 1.

FORCE
Specifies 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, national, or special characters. Enclose it in single quotation marks if it contains any special characters. A volume serial number is required.

Examples

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)
```

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 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

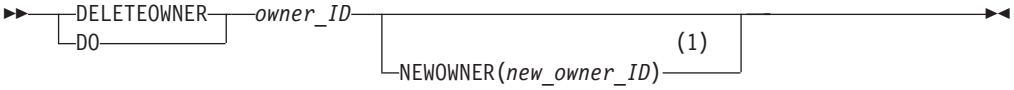
DELETEOWNER: Deleting Owner Information

Before you begin: To use the RMM DELETEOWNER subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

Use the DELETEOWNER subcommand 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 275 for more information.

Syntax



Notes:

1 The NEWOWNER operand must be specified to reassign volumes if the owner you are deleting owns one or more volumes.

Operands

NEWOWNER(*new_owner_ID***)**
Specifies 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
Specifies 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.

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)
```

Return Codes

See Chapter 10, "DFSMSrmm Return Codes and Reason Codes," on page 355 for DFSMSrmm reason codes.

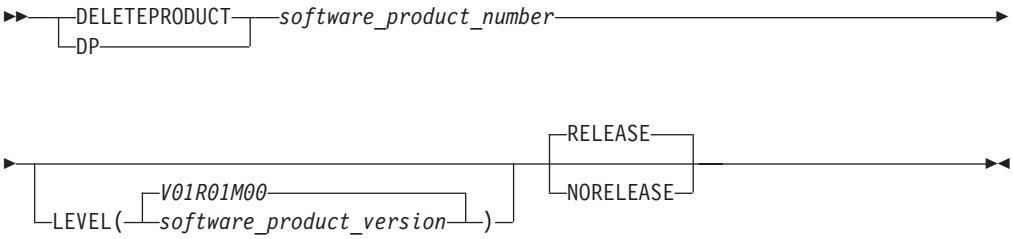
0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmm sets a reason code.
8	User not authorized.
12	Subcommand ended with an error. DFSMSrmm sets a reason code.
16	Error. The DFSMSrmm subsystem is not active.
20	Error. The data is incomplete or incorrect 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

Before you begin: To use the RMM DELETEPRODUCT subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

Use the DELETEPRODUCT subcommand 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.

Syntax



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
Specifies retaining 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 quotation marks if it contains any special characters or blanks. This operand is required and must immediately follow the DELETEPRODUCT subcommand.

RELEASE
Specifies releasing all volumes associated with the software product version.

RELEASE is the default value.

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
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin: To use the RMM DELETERACK subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.

This section describes the combined description for the DELETERACK subcommand and its alias DELETEBIN. See “DELETEBIN: Deleting Bin Number Information” on page 263 for information about using the RMM DELETEBIN subcommand alias.

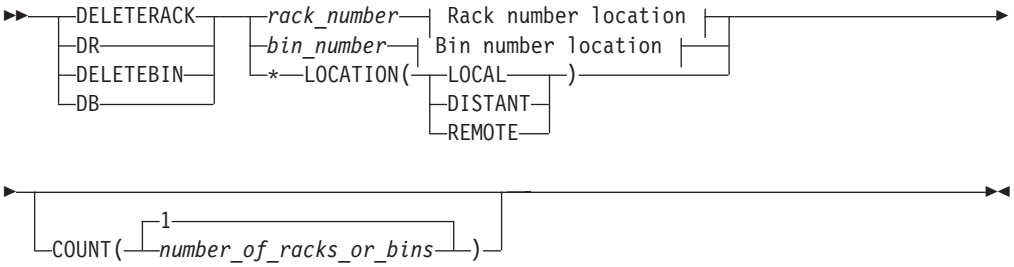
Use the DELETERACK subcommand to delete information about shelf locations you are no longer using in your removable media library or your storage locations. DFSMSrmm defines shelf space in the removable media library as rack numbers and bin numbers in storage locations.

See “DELETEBIN: Deleting Bin Number Information” on page 263 for information about using the RMM DELETERACK subcommand alias.

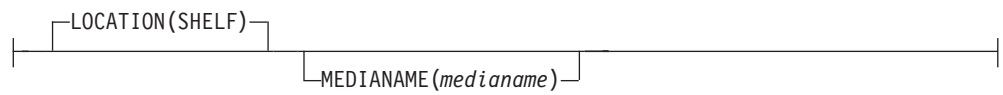
If you are deleting a specific rack number from the removable media library, you must specify a six-character rack number. If you are deleting more than one rack number, you must specify the initial rack number and a COUNT value. To delete rack numbers, DFSMSrmm uses the current VLPOOL definitions to 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 by starting with the highest bin number defined for the storage location 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 by starting with the initial bin number until it reaches the count value you specify or a bin number containing a volume.

Syntax



Rack number location:



Bin number location:

Operands

*bin_number|**

Specifies 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 DELETEBIN 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 the number of rack numbers that DFSMSrmm deletes from the removable media library or the number of 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 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***|*)**

Specifies 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 one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special 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 DELETEBIN subcommand.

Examples

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)
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

DELETEVOLUME: Deleting Volume Information

Before you begin:

- To use the RMM DELETEVOLUME subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.
- To use the RMM DELETEVOLUME REMOVE subcommand or the RMM DELETEVOLUME FORCE subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.
- To use the RMM DELETEVOLUME RELEASE subcommand,
  - You need READ access to the STGADMIN.EDG.RELEASE resource profile to release your own volumes.
  - When the STGADMIN.EDG.RELEASE resource profile is not defined, you need READ access to the STGADMIN.EDG.MASTER resource profile to release your own volumes.
  - If COMMANDAUTH(DSN) is in effect, you need READ access to the STGADMIN.EDG.MASTER resource profile or the STGADMIN.EDG.RELEASE resource profile. You need ALTER access to the first file data set name in the DATASET class, or if there is no first file defined to DFSMSrmm, you need ALTER access to the volume in the TAPEVOL class.

Related Reading:

- See “Deleting Information for Volumes in Multivolume Sets” on page 31 for information about using the DFSMSrmm ISPF dialog to delete volumes in a multi-volume set.
- See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

Use the DELETEVOLUME subcommand 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 the volume 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 that resides in a storage location or that is 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.

If you use the FORCE operand or the REMOVE operand 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 that reside in an automated tape library, the default is to eject the volume to a convenience

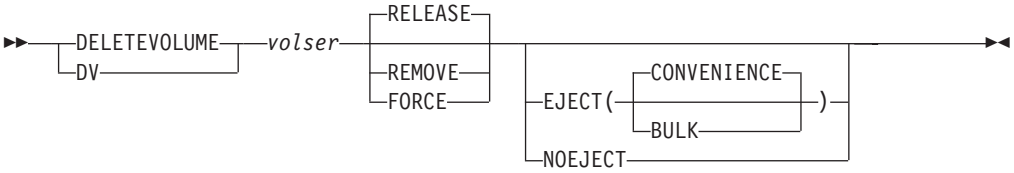
output station. You can also specify EJECT(BULK) to eject the volume to a high capacity output station. Specify the EJECT operand with the FORCE operand or the REMOVE operand. Specify NOEJECT with the FORCE operand or the REMOVE operand when you do not want to direct the volume to an exit station.

DFSMSrmm ignores the EJECT operand if you specify it for a volume that does not reside in a system-managed tape library or for a logical volume in private status. For a volume in scratch status, DFSMSrmm issues an eject, and the VTS Library Manager database entry for the volume might be purged.

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. DFSMSrmm also purges information for volumes that reside in a non-system-managed tape library that is considered part of a system-managed tape library and is defined in the TCDB as shelf resident.

See “Releasing Volumes” on page 102 for more information on releasing volumes.

Syntax



Operands

EJECT(CONVENIENCE|BULK)
Specifies directing 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
Specifies deleting 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.

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. You can use FORCE to delete an empty stacked volume. Prior to using the FORCE operand to delete an empty stacked volume, you must remove all the contained volumes in a stacked volume using the RMM CHANGEVOLUME subcommand for each volume in the stacked volume container.

FORCE is mutually exclusive with RELEASE and REMOVE. RELEASE is the default.

NOEJECT
Specifies preventing 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
Specifies releasing the volume according to the release actions set for the volume. You can use RELEASE for logical and physical volumes only.

RELEASE is mutually exclusive with FORCE and REMOVE. RELEASE is the default.

REMOVE
Specifies deleting information recorded by DFSMSrmm about a scratch volume that is no longer wanted, and changing 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 use REMOVE to delete an empty stacked volume. Prior to using the REMOVE operand to delete an empty stacked volume, you must remove all the contained volumes in a stacked volume using the RMM CHANGEVOLUME subcommand for each volume in the stacked volume container.

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, national, or special characters. Enclose it in single quotation marks if it contains any special characters. You can only specify a particular volume serial. A volume serial number is required and must follow the DELETEVOLUME subcommand.

Example

Task: Delete information about the scratch volume with volume serial number 8E1U01.

Command:

```
RMM DELETEVOLUME 8E1U01 REMOVE
```

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

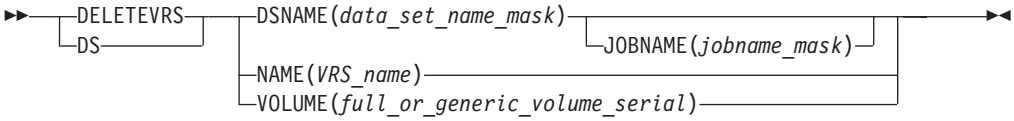
DELETEVRS: Deleting Vital Record Specifications

Before you begin: To use the RMM DELETEVRS subcommand, you need CONTROL access to the STGADMIN.EDG.VRS resource profile to delete vital record specifications.

Use the DELETEVRS subcommand 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.

Syntax



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 205 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')`.

DSNAME is mutually exclusive with the NAME and VOLUME operands.

JOBNAME(*jobname_mask***)**
Specifies a specific jobname or a jobname mask. A job name is one-to-eight alphanumeric characters or $, #, or @. Job name must start with an alphabetic character, $, #, or @. 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, national, or special characters. A generic volume serial number is one-to-five alphanumeric, national, or special characters followed by an asterisk. Enclose it in single quotation marks if it contains any special characters.

VOLUME is mutually exclusive with the DSNAME and NAME operands.

Examples

Task: Delete the volume vital record specification for the volume 8E1U02.

Command:

```
RMM DELETEVRS VOLUME(8E1U02)
```

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 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

GETVOLUME: Requesting and Assigning Scratch Volumes

Before you begin: To use the RMM GETVOLUME subcommand:

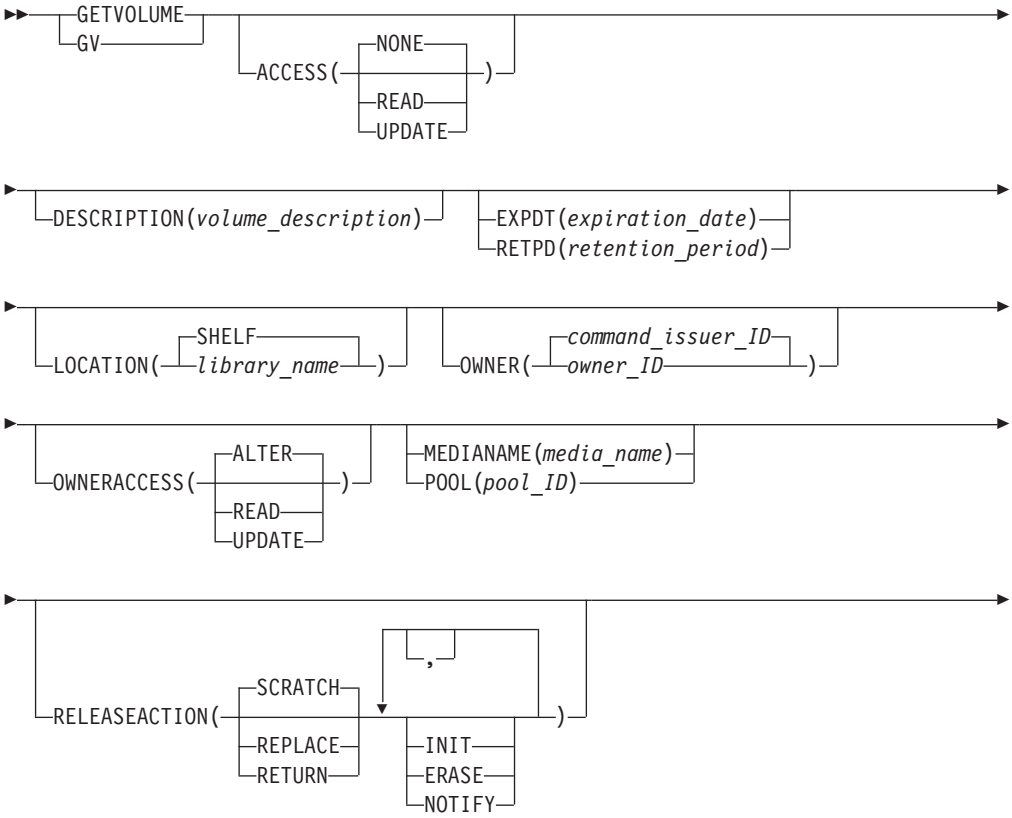
- You need READ access to the STGADMIN.EDG.MASTER resource profile to request a volume for yourself.
- You need CONTROL access to the STGADMIN.EDG.MASTER resource profile to request volumes for any user

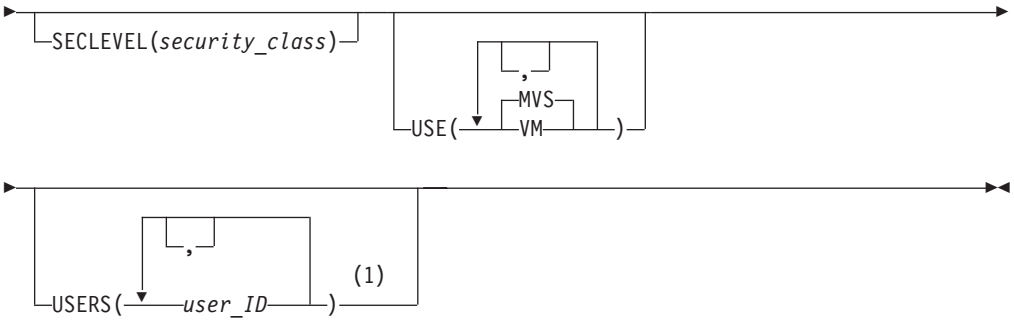
Use the GETVOLUME subcommand to request a scratch volume and assign it to an owner defined to DFSMSrmm. The default owner is the owner who issues the command.

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.

When DFSMSrmm assigns a volume, it sets the volume status to USER which means that the volume can be overwritten by any data set. DFSMSrmm prevents you from reading any existing data on a scratch volume that is obtained using the GETVOLUME subcommand.

Syntax





Notes:

1 You can specify a maximum of 12 user IDs.

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 quotation marks if it contains any special characters or blanks.

EXPDT(*expiration_date*)
Specifies date the volume should be considered for release.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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. You cannot specify a distributed library name.

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 287 for more information.

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 alphanumeric, national, or special characters followed by an asterisk. Enclose it in single quotation marks if it contains any special characters. If you do not specify a pool ID, DFSMSrmm assigns a volume from the default scratch pool.

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 287 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 287 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 volume as defined by the ACCESS keyword. You can specify a maximum of twelve user IDs separated by blanks or commas.

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)
```

Return Codes

See Chapter 10, "DFSMSrmm Return Codes and Reason Codes," on page 355 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. The data is incomplete or incorrect 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

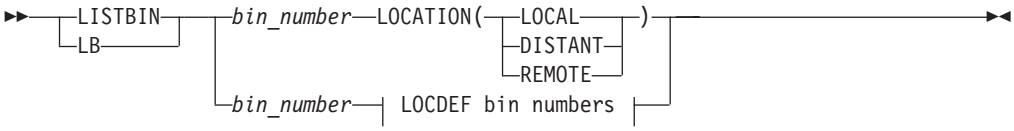
Before you begin: To use the RMM LISTBIN subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

The LISTBIN subcommand is an alias for the LISTRACK subcommand. See "LISTRACK: Displaying Information about a Shelf Location" on page 302 for the combined description of the LISTRACK and LISTBIN operands.

Use the LISTBIN subcommand 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 that are defined to DFSMSrmm. See "SEARCHBIN: Creating a List of Bin Numbers" on page 310 for more information.

Syntax



LOCDEF bin numbers:

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.

LOCATION(LOCAL|DISTANT|REMOTE|*LOCDEF_location_name***)**
Specifies 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***|*)**
Specifies the media for the bins that are displayed. MEDIANAME is required when you request a display of an installation defined storage location.

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 70:

```
Rack/bin number = 123456      Location =             Medianame =
Status          =                                    Pool      =
Volume:
  Current       =
  Moving-in     =
  Moving-out    =
  Old           =
```

Figure 70. Sample LISTBIN Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

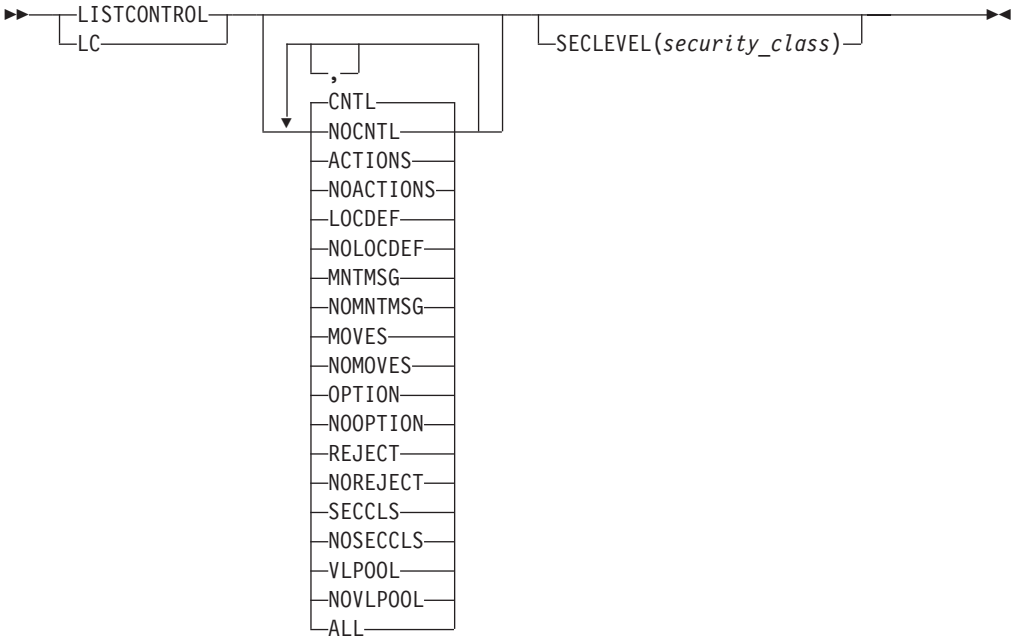
Before you begin: To use the RMM LISTCONTROL subcommand:

- You need CONTROL access to the STGADMIN.EDG.MASTER resource profile when the STGADMIN.EDG.LISTCONTROL resource profile is not defined.
- You need CONTROL access to the STGADMIN.EDG.LISTCONTROL resource profile when the STGADMIN.EDG.LISTCONTROL resource profile is defined.

Use the LISTCONTROL subcommand to display information in the control record of the control data set and information that is defined in the EDGRMM*xx* parmlib member. Figure 71 on page 290 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.

When you issue the LISTCONTROL CNTL ACTIONS MOVES command when you are running DFSMSrmm client/server support, DFSMSrmm returns information from the control data set on the DFSMSrmm server system. When you issue all other LISTCONTROL commands, DFSMSrmm returns information from the system where the command is issued.

Syntax



Operands

ACTIONS Specifies 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 Specifies 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 Specifies 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.

LOCDEF Specifies 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 Specifies 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 Specifies to display the mount/fetch message definitions that are currently in use.

NOACTIONS Specifies to prevent outstanding volume actions from being listed when you specify LISTCONTROL ALL.

NOCNTL Specifies to prevent the control record information from being displayed when you specify LISTCONTROL ALL.

NOLOCDEF Specifies to prevent the location definition information from being displayed when you specify LISTCONTROL ALL.

NOMOVES Specifies to prevent outstanding volume moves from being listed when you specify LISTCONTROL ALL.

NOMNTMSG Specifies to prevent the mount/fetch message definitions from being displayed when you specify LISTCONTROL ALL.

NOOPTION Specifies to prevent system options from being displayed when you specify LISTCONTROL ALL.

NOREJECT Specifies to prevent volumes that are unavailable on the system from being displayed when you specify LISTCONTROL ALL.

NOSECCLS Specifies to prevent security classes from being displayed when you specify LISTCONTROL ALL.

NOVLPOOL Specifies to prevent volume pool details from being displayed when you specify LISTCONTROL ALL.

OPTION Specifies to display the system options that are currently in use.

REJECT Specifies to display rack number and access information about volumes that are unavailable on the system.

SECCLS Specifies to display the security classes defined for your installation.

SECLEVEL(*security_class***)**
Specifies 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 Specifies to display information about pools defined for your installation.

Examples

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 71 on page 290:

```
Control record:
Type = MASTER    Create date = 2004/007    Create time = 05:18:41
Journal: Utilization =   0% ( 0% threshold)     STATUS: = ENABLED
CDS:     Utilization =   0%
Exit status:                              Options:
  EDGUX100 = NONE                           Stacked Volumes         = NONE
  EDGUX200 = NONE                           Extended Bin            = ENABLED
Last backup:                              Last expiration processing:
  Date =             Time =                 Date = 2004/019    Time = 05:05:33
Last journal backup:                      Last store update:
  Date =             Time =                 Date =             Time =
Last report extract:                      Last VRS processing:
  Date =             Time =                 Date = 2004/019    Time = 05:05:33
Last scratch procedure:                   Last Catalog synchronize:
  Date =             Time =                 Date = 2004/013    Time = 05:04:57
Rack numbers       = 1                    Empty racks        = 0
LOCAL store bins   = 0                    Empty LOCAL bins   = 0
DISTANT store bins = 0                    Empty DISTANT bins = 0
REMOTE store bins  = 0                    Empty REMOTE bins  = 0
Control functions in progress:
Backup         = N  Restore        = N
Verify         = N  Expiration     = N
Report Extract = N  Disaster Store = N
VRS            = N  Synchronize    = N
Client/Server:
  host name  = EZU34.TUCSON.IBM.COM
  IP address = 9.11.194.222


System options:
PARMLIB Suffix  = C1
Operating mode  = P    Retention period: Default = 6       Maximum = 6666
                                         Catalog = 12    hours
Control data set name      =
Journal file data set name =
Journal threshold          =  0%
Catalog SYSID              = Set
Scratch procedure name     = S1ATL008
Backup procedure name      =
IPL date check  = N        Date format    = J       RACF support   = P
SMF audit       = 0        SMF security   = 249     CDS id         = CSTEST
MAXHOLD value   = 250      Lines per page = 54      System ID      = EZU34
BLP             = RMM      TVEXT purge    = RELEASE Notify         = Y
Uncatalog       = Y        VRS job name   = 2       Message case   = M
MASTER overwrite= LAST     Accounting     = J       VRS selection  = OLD
VRS change      = INFO     VRSMIN action  = FAIL    VRSMIN count   = 1
Disp DD name    =          Disp msg ID    = EDG4054I
Retain by       = VOLUME   Move by        = VOLUME  CMDAUTH Owner  = YES
PREACS          = NO       SMSACS         = NO      CMDAUTH Dsn    = NO
Reuse bin       = STARTMOVE                         Media name     = 3480
                                                    Local tasks    = 5
PDA: ON
 Block count    = 100      Block size     = 22      Log            = ON
SMSTAPE:
 Update scratch = YES      Update command = YES     Update exits   = YES
 Purge          = YES
Client/Server:
 Subsystem type = CLIENT   Port           = 4021
 Server                    Server tasks   = 0
  host name  = EZU35.TUCSON.IBM.COM
  IP address = 9.11.194.223
```

Figure 71. Sample LISTCONTROL Output

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 72.

```
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 Ty Expdt Pool     Media    Description
                CF pe check name     name
------ -------- -- -- ----- -------- -------- -------------------------
                     Action Scratch  Overwrite
                     ------ -------  ---------
*      BTLS110  N  S  N     SCRTCH2  3490     BTLS POOL
                            AUTO     LAST
P4L33*          N  S  O              TAPE     DIALOGUE VOLUME TESTS
                            MANUAL   ADD
P4L41*          N  S  O              TAPE     DIALOGUE VOLUME TESTS 1
                            AUTO     MATCH
P4L42*          N  R  O              TAPE     DIALOGUE VOLUME TESTS 2
                            AUTO     USER
P4L43*          N  S  O              TAPE     DIALOGUE VOLUME TESTS 3
                            AUTO     LAST
R2*             N  R  O              TAPE     DIALOGUE RACK TESTS
                            AUTO     USER
R5*             N  R  O              TAPE     DIALOGUE PRODUCT TESTS
                            AUTO     USER
S3*             N  S  O              TAPE     DIALOGUE DATASET TESTS
                            AUTO     USER
TE520*          Y  S  O              3420     TEST T14000 3420S
                            AUTO     LAST
T1*             N  R  O              3480     DIALOGUE OWNER TESTS
                            AUTO     LAST
T3*             N  R  O              TAPE     DIALOGUE DATASET TESTS
                            AUTO     LAST
T4*             N  R  O              TAPE     DIALOGUE VOLUME TESTS
                            AUTO     LAST
1I501*          Y  S  O              3480     TEST T12000 SPECIFIC
                            AUTO     MATCH
*               N  S  O              3480     DEFAULT POOL
                            AUTO
```

Figure 72. Sample LISTCONTROL Output

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 73:

```
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 73. Sample LISTCONTROL Output

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 74.

```
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 74. LISTCONTROL LOCDEF Output - No LOCDEFs Defined

Table 32 describes the columns in the LISTCONTROL subcommand output.

Table 32. LISTCONTROL Subcommand Output

Column	Description
Location	The location name, in ascending alphanumeric 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. If defined as a storage home location it shows HSTORE.
Priority	Is the current relative priority of the location.
Medianames	Provides all the media names specified for the location.

Figure 75 on page 294 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
SITEX     Y  BINS    HSTORE   3500   CARTS,*
```

Figure 75. LISTCONTROL LOCDEF Output - LOCDEFs Defined

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

LISTDATASET: Displaying Information about a Data Set

Before you begin: To use the LISTDATASET subcommand:

- You need READ access to the STGADMIN.EDG.MASTER resource profile.
- In addition, if COMMANDAUTH(DSN) is in effect, you need READ access to the data set name in the DATASET class.

Related Reading: See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

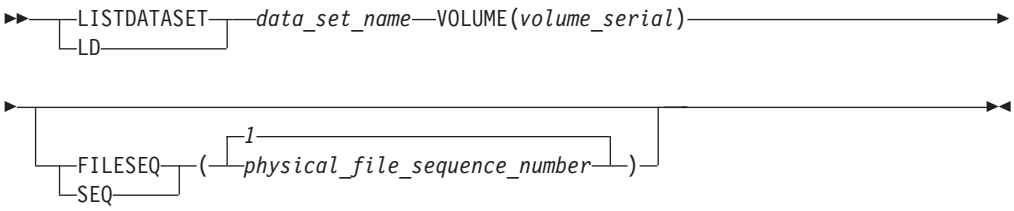
Use the LISTDATASET subcommand to display information about a single data set defined to DFSMSrmm. You must specify the data set name and the volume serial number where the data set resides. If the data set is not the first data set on the volume, you must also specify a sequence number.

The output from the LISTDATASET subcommand includes:

- The creating job name
- Data set retention date
- Storage group
- Storage class
- Data class
- The matching VRS type:
  - DATASET type indicates that the data set is retained by a DSNAME type vital record specification.
  - SMSMC type indicates that the data set is retained by a vital record specification that matches its SMS management class.
  - VRSMV type indicates that the data set is retained by a vital record specification that matches its VRS management value.
  - DSN/MV type indicates that the data set is retained by a DSNAME type vital record specification and a management value VRS defined with WHILECATALOG.
- The 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 315 for more information.

Syntax



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 65535.

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, national, or special characters. Enclose it in single quotation marks if it contains any special characters. This operand is required.

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 76:

```
Data set name = NISHINAL.TEST.ORDER
Volume        = BC0047                Physical file sequence number = 1
Owner         = RMMUSER                        Data set sequence = 1
Create date   = 12/20/2002 Create time = 06:16:06 System ID         = W98MVS1
Expiration date       = 2003/02/16 Original Expir. Date  = 2003/10/27
Block size            = 80         Block count           = 10
Percent of volume     = 0          Total block count     = 10
Logical Record Length = 80         Record Format         = FB
Date last written     = 12/20/2002 Date last read        = 12/20/2002
Job name              = RMMUSERJ   Last job name         = RMMUSERJ
Step name             = WRITE1     Last step name        = READ1
Program name          = TAPEIO     Last program name     = IEBGENER
DD name               = TAPE       Last DD name          = SYSUT1
Device number         = 0911       Last Device number    = 0910
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   =           Description   =
```

Figure 76. Sample LISTDATASET Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin: To use the RMM LISTOWNER subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

Use the LISTOWNER subcommand to display information about a single owner defined to DFSMSrmm.

Syntax

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.

Example

Task: Request information recorded by DFSMSrmm about the owner whose owner ID is D023425.

Command:

```
RMM LISTOWNER D023425
```

Output: DFSMSrmm displays information such as that shown in Figure 77.

```
Owner       = D023425
Last name   = SEYFERT                First names = CLEMENS
Department  = 4193
Address     = IBM
              HECHTSHEIMER STRASSE
              55131 MAINZ - GERMANY
Telephone:
Internal    = 1579                   External    = 02137-841579
Electronic mail:
Userid      = CSEYFERT               Node        = IBMDE

Volumes     = 18
```

Figure 77. Sample LISTOWNER Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

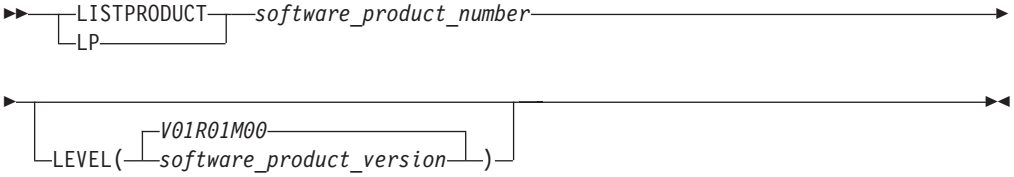
LISTPRODUCT: Displaying Information about a Software Product

Before you begin: To use the RMM LISTPRODUCT subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

Use the LISTPRODUCT subcommand 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 that are defined to DFSMSrmm. See “SEARCHPRODUCT: Creating a List of Software Products” on page 322 for more information.

Syntax



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 quotation marks if it contains any special characters or blanks. This operand is required and must immediately follow the LISTPRODUCT subcommand.

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 78 on page 301.

```
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 78. Sample LISTPRODUCT Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

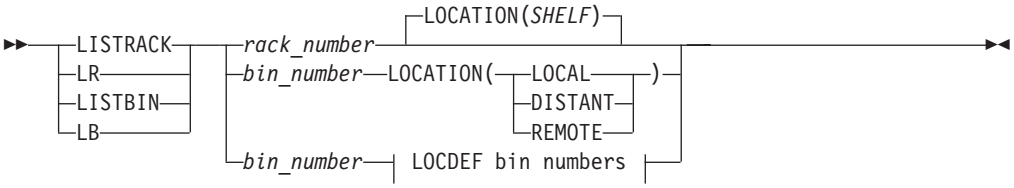
Before you begin: To use the RMM LISTRACK subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

This section describes the combined description for the LISTRACK subcommand and its alias LISTBIN. See “LISTBIN: Displaying Information about a Shelf Location” on page 285 for information about using the RMM LISTRACK subcommand alias.

Use the LISTRACK subcommand 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 storage location or in an installation-defined storage location.

Use the SEARCHRACK subcommand or SEARCHBIN subcommand to request lists of rack numbers or bin numbers that are defined to DFSMSrmm. See “SEARCHRACK: Creating a List of Shelf Locations” on page 325 and “SEARCHBIN: Creating a List of Bin Numbers” on page 310 for more information.

Syntax



LOCDEF bin numbers:

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 LISTRACK 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***|*)**
Specifies 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 one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. You cannot use a generic rack number.

A rack or bin number is required. The rack or bin number must immediately follow the LISTRACK or LISTBIN subcommand.

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 79: :

```
Rack/bin number = 123456      Location =             Media name =
Status                                               Pool       =
Volume:
  Current       =
  Moving-in     =
  Moving-out    =
  Old           =
```

Figure 79. Sample LISTRACK Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin: To use the RMM LISTVOLUME subcommand:

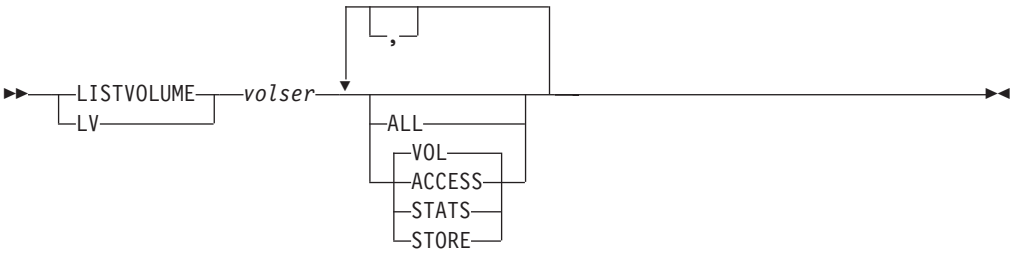
- You need READ access to the STGADMIN.EDG.MASTER resource profile.
- In addition, if COMMANDAUTH(DSN) is in effect, you need READ access to the first file data set name in the DATASET class. If there is no first file defined to DFSMSrmm and the volume is in master status or user status, you need READ access to the volume in the TAPEVOL class.

Related Reading: See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

Use the LISTVOLUME subcommand to display information that is recorded by DFSMSrmm for a single volume. You must specify a volume serial number. You can optionally limit the amount of information DFSMSrmm displays.

Use the SEARCHVOLUME subcommand to create lists of volumes that are defined to DFSMSrmm. See "SEARCHVOLUME: Creating a List of Volumes" on page 329 for more information.

Syntax



Operands

ACCESS Specifies the access information for the volume. Access information includes owner and user authorization, last change information, access list, and operating system usage.

ALL Specifies all the information recorded by DFSMSrmm for the volume. Specifying ALL is equivalent to specifying the operands VOL, ACCESS, STATS and STORE.

STATS Specifies 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 Specifies the storage information for the volume. Storage location information includes the current location of a volume, movement information, store data and bin numbers.

VOL Specifies 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 serial number. A fully qualified serial number is

one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. You can only specify a particular volume serial. A volume serial number is required and must follow the LISTVOLUME subcommand.

Examples

Task: Request the volume serial information and access for the volume with serial number 003186.

Command:

```
RMM LISTVOLUME 003186 VOL ACCESS
```

Output: DFSMSrmm displays information such as that shown in Figure 80:

```
Volume Information:
Volume = 003186   VOL1  = INT001   Rack  = 003186    Owner   = RMMUSER
       Type = PHYSICAL         Stacked count  = 0           Jobname =
Creation: Date  = 10/01/2000 Time          = 05:04:00  System ID = SYSA
Assign:   Date  = 11/01/2000 Time          = 05:15:00  System ID = SYSB
Expiration Date = 2002/029   Original      =
Retention Date  =
Data set name =
Volume Status:
Status = MASTER    Availability =                  Label = SL
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 =

Access information:
Owner access = ALTER   Volume access = NONE    Last change = D027182
VM use = N   MVS use = Y
Access list:
RMMUSER   IBMUSER
```

Figure 80. Sample LISTVOLUME Output

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 81 on page 306:

```
Statistics:
Number of data sets = 32768   Data set recording= ON
Volume usage(Kb)= 0           Use count         = 0
Volume capacity = 0           Percent full      = 0
Date last read  = 11/18/2002  Date last written = 11/18/2002
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:
Movement tracking date =                Intransit    =  N
In container           =                Move mode    =  AUTO

Location:   Current     Destination   Old       Required   Home
Name=       SHELF                                          SHELF
Type=
Bin number=
Media name
```

Figure 81. Sample LISTVOLUME Output

Notes:

1. 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. The Volume usage is an approximation of the number of bytes written by the application.
2. The Use count field displays the number of times that the volume was opened for either read or write operations. DFSMSrmm resets the volume use count field to zero if the volume status is changed from SCRATCH to MASTER or USER.

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

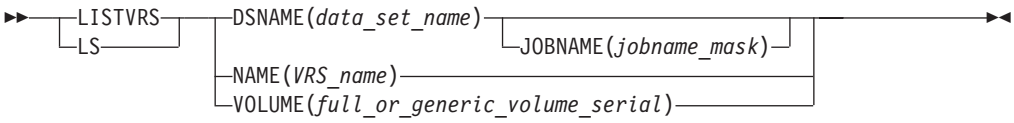
Before you begin: To use the RMM LISTVRS subcommand, you need READ access to the STGADMIN.EDG.VRS resource profile.

Use the LISTVRS subcommand 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 347 for more information.

Syntax



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').

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, national, or special characters. A generic volume serial number is one-to-five alphanumeric, national, or special characters followed by an asterisk. Enclose a full or generic volume serial number in single quotation marks if it contains any special characters. You can specify a generic volume serial number if it exactly matches an existing vital record specification.

VOLUME cannot be used with the DSNAME and NAME operands.

Examples

Task: List a vital record specification based on job name by coding both DSNAME and JOBNAME where 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 82:

```
RMM ADDVRS DSNAME('A.B') WHILECATALOG
RMM ADDVRS DSNAME('A.B') JOBNAME(BENSJOY) WHILECATALOG
```

Figure 82. Listing Vital Record Specification Information Examples

Command:

```
RMM LISTVRS DSNAME('A.B')
```

DFSMSrmm does not list the second vital record specification. You must also specify the JOBNAME operand on the LISTVRS subcommand.

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 83 on page 309:

```
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    = AADN1     using ANDVRS

VRS Owner    = OWN000
Description  = User data sets
Vital Record Specification to be deleted on 31/12/1999
```

Figure 83. Sample LISTVRS Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHBIN: Creating a List of Bin Numbers

Before you begin: To use the RMM SEARCHBIN subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

Use the SEARCHBIN subcommand as shown in Figure 84 on page 311 to create a list of shelf locations that are defined to DFSMSrmm. DFSMSrmm defines shelf space in storage locations as bin numbers.

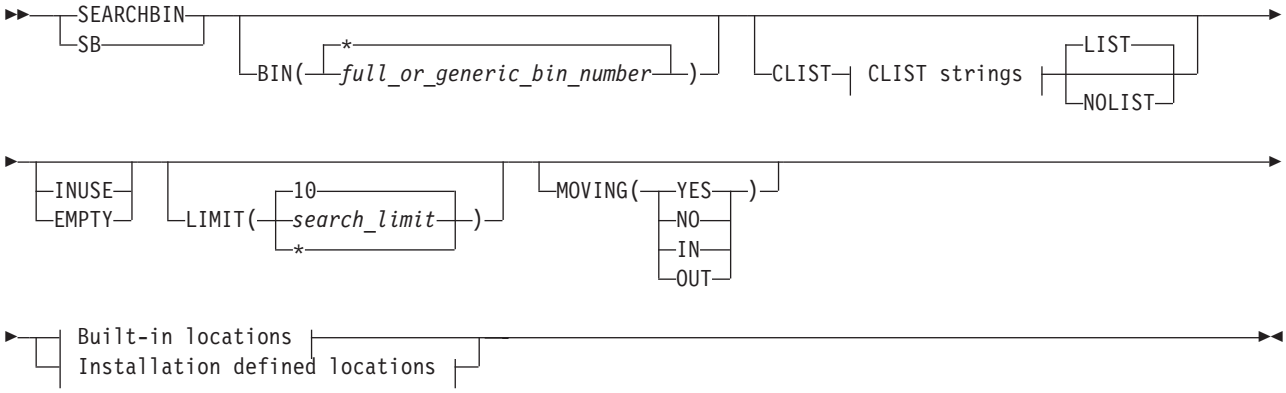
You can restrict how many bin numbers DFSMSrmm lists by specifying the LIMIT operand. DFSMSrmm searches until your limit is reached or until it lists all shelf locations that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten shelf locations.

Table 33 shows the information DFSMSrmm returns for each bin number in the list, in the order it is displayed:

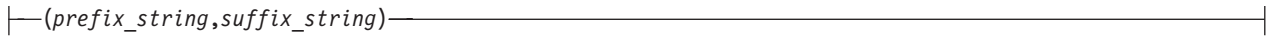
Table 33. Information Returned by SEARCHBIN

Table Field Name	Description
Rack/Bin	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
Moving-in-volume	Volume that will move into this bin
Moving-out-volume	Volume that will move out of this bin
Old volume	Volume that was previously in this bin

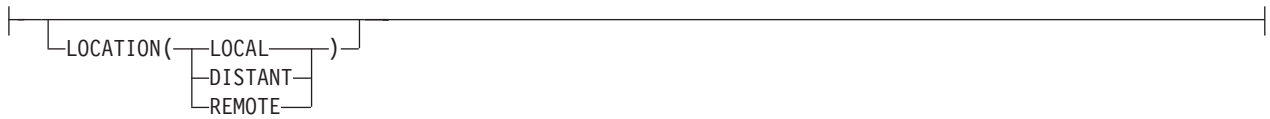
Syntax



CLIST strings:



Built-in locations:



Installation defined locations:

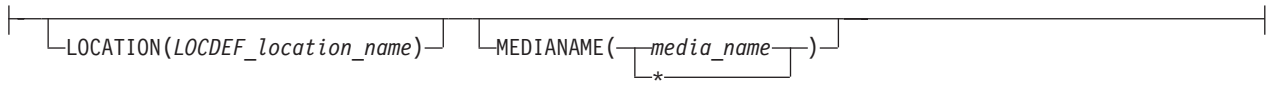


Figure 84. SEARCHBIN Syntax Diagram

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 a CLIST to create a data set of executable commands. You can edit the data set to remove any bin numbers you do not want in the list. Then you can run the CLIST at your convenience.

DFSMSrmm returns the number of the shelf location for each record if you do not specify (*prefix_string* and *suffix_string*).

You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). These text

strings cannot exceed 255 characters. Separate the *prefix_string* and *suffix_string* using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns.

See "Creating CLISTs of Executable Subcommands" on page 128 for information about the data set used for the CLIST output.

EMPTY
Specifies bins in a storage location which are available for use.

INUSE
Specifies bins in a storage location that are occupied by a volume and not available for another volume.

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***)**
Specifies a 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***l*)**
Specifies that the list is limited to shelf locations containing volumes belonging to the same media name. The 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. 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 287 for more information.

MOVING(YES|NO|IN|OUT)
Specify to list bins based on the moving status of their assigned volumes.

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.

Example

Task: Create a list of empty bin numbers that are available for use in the DPBINS storage location.

Command:

```
RMM SEARCHBIN BIN(A*) LOCATION(DPBINS) MEDIANAME(3480) EMPTY
```

Output: DFSMSrmm displays a list such as the one shown in Figure 85.

```
Rack/Bin          Media             Current  Moving-in  Moving-out  Old
number   Location  name      Status  volume   volume     volume      volume
-------- --------  --------  ------- -------  ---------  ----------  ------
A00001   DPBINS    3480      EMPTY
A00002   DPBINS    3480      EMPTY
A00003   DPBINS    3480      EMPTY
A00004   DPBINS    3480      EMPTY
A00005   DPBINS    3480      EMPTY
A00006   DPBINS    3480      EMPTY
A00007   DPBINS    3480      EMPTY
A00008   DPBINS    3480      EMPTY
A00009   DPBINS    3480      EMPTY
A00010   DPBINS    3480      EMPTY
EDG3203I SEARCH COMPLETE - MORE ENTRIES MAY EXIST
EDG3012I 10         ENTRIES LISTED
```

Figure 85. Sample SEARCHBIN Output for Empty Bins

Task: Create a list of bin numbers.

Command:

```
RMM SEARCHBIN BIN(*)
```

Output: DFSMSrmm displays a list such as the one shown in Figure 86.

```
Rack/Bin          Media             Current  Moving-in  Moving-out  Old
number   Location  name      Status  volume   volume     volume      volume
-------- --------  --------  ------- -------  ---------  ----------  ------
000001   LOCAL               IN USE   A09999
000002   LOCAL               EMPTY
000003   LOCAL               IN USE   A09003
000004   LOCAL               EMPTY    A09004
000005   LOCAL               EMPTY    A09003
000006   LOCAL               IN USE   A09005
000007   LOCAL               EMPTY
000008   LOCAL               EMPTY
000009   LOCAL               EMPTY
000010   LOCAL               EMPTY
EDG3203I SEARCH COMPLETE - MORE ENTRIES MAY EXIST
EDG3012I 10         ENTRIES LISTED
```

Figure 86. Sample SEARCHBIN Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHDATASET: Creating a List of Data Sets

Before you begin: To use the RMM SEARCHDATASET subcommand:

- You need READ access to the STGADMIN.EDG.MASTER resource profile.
- In addition, when COMMANDAUTH(DSN) is in effect, you need READ access to the data set name in the DATASET class.

Related Reading: See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

Use the SEARCHDATASET subcommand 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 that are created by a particular job name.

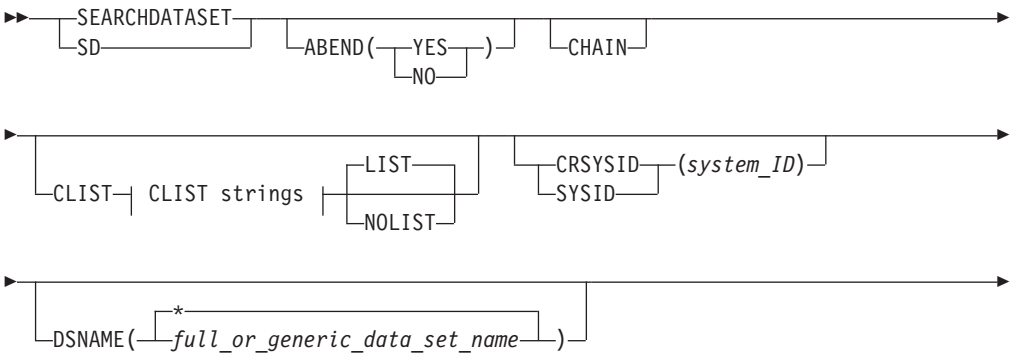
You can restrict how many data sets DFSMSrmm lists by specifying the LIMIT operand. DFSMSrmm searches until it reaches your limit or until it finds all data sets that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten data sets.

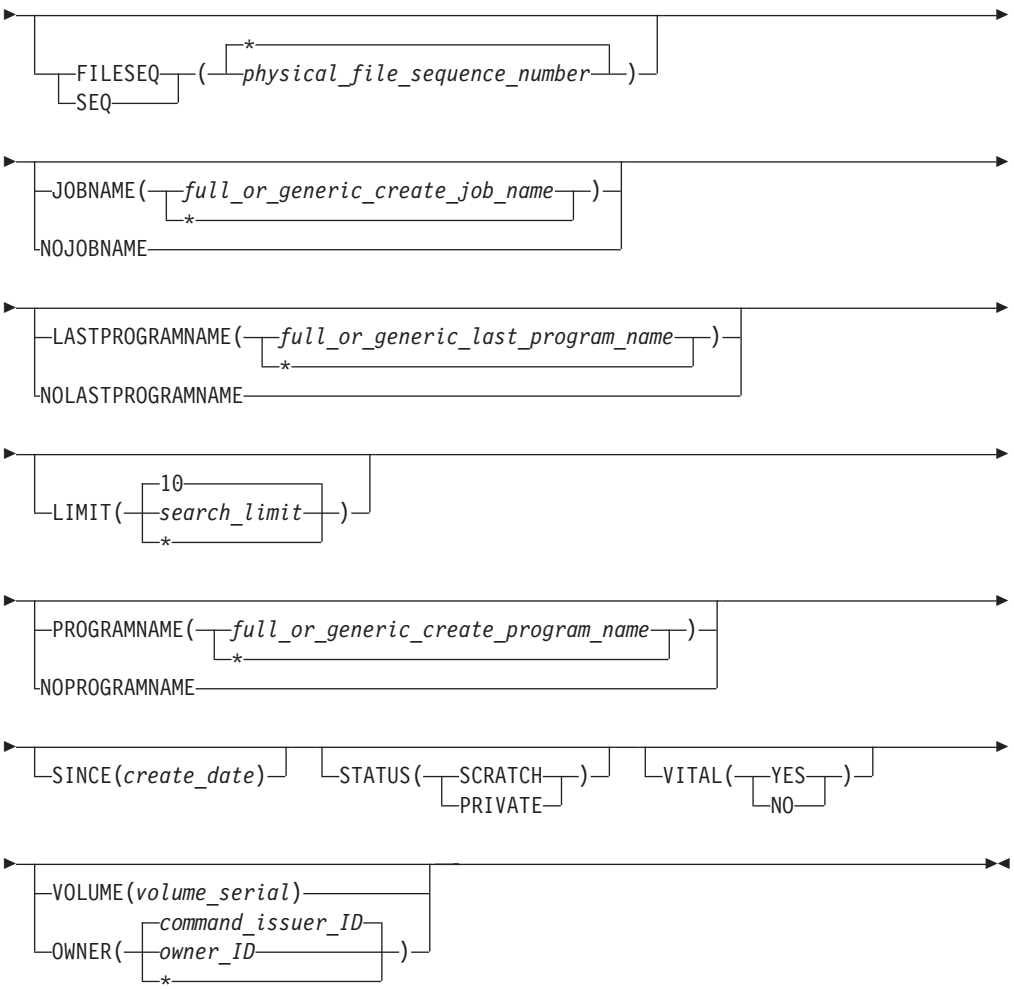
Table 34 shows the information DFSMSrmm returns for each data set in the order it is displayed:

Table 34. Information Returned by SEARCHDATASET

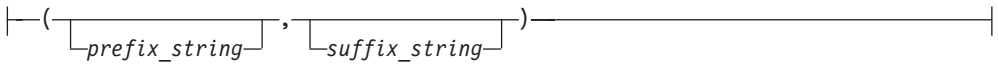
Table Field Name	Description
Data set name	Data set name
Volume	Volume serial number of the volume on which the data set resides
Owner	Owner ID of the volume owner
Create date	Date the data set was created
Seq	Physical file sequence number

Syntax





CLIST strings:



Operands

ABEND(YES|NO)
Specifies the search to volumes containing data sets that were closed as a result of ABEND processing.

CHAIN
Specifies the 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***)**
Specifies a CLIST to create a data set of executable commands. You can edit the data set to remove any data sets you do not want in the list. Then you can run the CLIST at your convenience.

DFSMSrmm returns the name of the data set, a data set sequence number, and the volume serial number of the volume on which the data set resides you do not specify (*prefix_string* and *suffix_string*).

You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). These text strings cannot exceed 255 characters. Separate the *prefix_string* and *suffix_string* using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns.

See “Creating CLISTs of Executable Subcommands” on page 128 for information about the data set used for the CLIST output.

CRSYSID(*creating_system_ID***)**
Specifies data sets based on the ID of the system on which the data set was created. Specify a one-to-eight character unique system name.

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.

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(*|*physical_file_sequence_number***)**
Specifies data sets based on the relative position on a volume. The maximum allowable decimal value is 65535. Specify an asterisk (*) 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 @ and must start with an alphabetic

character, $, #, or @. You can use a generic jobname. Use % in your generic jobname mask to match any one character and * to match any character string in the jobname.

If you do not specify JOBNAME, all data sets are listed.

If you specify JOBNAME(*), DFSMSrmm returns all data sets that match the specified data set name that have a job name. Data sets that do not have a job name are not listed. JOBNAME is mutually exclusive with NOJOBNAME.

LASTPROGRAMNAME(*full_or_generic_last_program_name***)**
Specifies data sets by the last referenced program name. A program name is one-to-eight alphanumeric characters, $, #, or @. The program name must start with an alphabetic character, $, #, or @. Use % in your generic program name mask to match any one character and * to match any character string in the program name. If you specify LASTPROGRAMNAME(*), DFSMSrmm returns all the data sets that have a last referenced program name. If you do not specify LASTPROGRAMNAME or NOLASTPROGRAMNAME, DFSMSrmm returns all data sets. LASTPROGRAMNAME is mutually exclusive with NOLASTPROGRAMNAME.

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.

NOJOBNAME
Specifies data sets that do not have a job 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.

NOLASTPROGRAMNAME
Specifies to select data sets that do not have a last referenced program name. NOLASTPROGRAMNAME is mutually exclusive with LASTPROGRAMNAME.

NOPROGRAMNAME
Specifies to select data sets that do not have a creating program name. NOPROGRAMNAME is mutually exclusive with PROGRAMNAME.

PROGRAMNAME(*full_or_generic_create_program_name***)**
Specifies the PROGRAMNAME operand to search for data sets with a creating program name recorded. DFSMSrmm records the name of the job step program running at the time the data set is opened for output. A program name is one-to-eight alphanumeric characters, $, #, or @. The program name must start with an alphabetic character, $, #, or @. Use % in your generic program name mask to match any one character and * to match any character string in the program name. If you specify PROGRAMNAME(*), DFSMSrmm returns all the data sets that have a program name. Use the NOPROGRAMNAME operand to search for data sets with no creating data set name recorded. If you do not specify PROGRAMNAME or NOPROGRAMNAME, DFSMSrmm returns all data sets. PROGRAMNAME is mutually exclusive with NOPROGRAMNAME.

OWNER(*user_ID*|*owner*|***)**
Specifies 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.

If you specify both the OWNER operand and the VOLUME operand of this subcommand, DFSMSrmm processes the last operand specified. 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***)**
Specifies that the SINCE operand with a create date lists data sets created after the specified date. DFSMSrmm returns a list of data sets that were created after the create date that you specify. Specify the year and day in the create date in one of two forms:

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001.

If you do not specify the SINCE parameter, DFSMSrmm considers all data sets regardless of the date when they were created.

STATUS(SCRATCH|PRIVATE)
Specifies 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)
Specifies 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***)**
Specifies the search to data sets residing on the indicated volume. A volume serial is one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters.

If you do not specify a volume serial, DFSMSrmm considers all volumes.

If you specify both the VOLUME operand and the OWNER operand of this subcommand, DFSMSrmm processes the last operand specified. If you do not specify VOLUME or OWNER, DFSMSrmm uses the user ID of the command issuer as the default OWNER ID.

Examples

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 87:

```
Data set name                               Volume Owner    Create date Seq
------------------------------------------- ------ -------- ----------- ----
DATA.SET.FIVE                               VOL001 OWN000   31/03/2002  1
DATA.SET.FOUR                               VOL000 OWN000   21/03/2002  4
DATA.SET.SEVEN                              VOL002 OWN000   16/03/2002  1
DATA.SET.TWO                                VOL000 OWN000   17/03/2002  2
EDG3012I 4          ENTRIES LISTED
```

Figure 87. Sample SEARCHDATASET Output

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 88:

```
Data set name                               Volume Owner    Create date Seq
------------------------------------------- ------ -------- ----------- ----
DATA.SET.ONE                                VOL000 OWN000   19/02/2002  1
DATA.SET.TWO                                VOL000 OWN000   01/03/2002  2
DATA.SET.THREE                              VOL000 OWN000   09/02/2002  3
DATA.SET.FOUR                               VOL000 OWN000   21/03/2002  4
EDG3012I 4          ENTRIES LISTED
```

Figure 88. 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 89:

```
Data set name                               Volume Owner    Create date Seq
------------------------------------------- ------ -------- ----------- ----
DATA.SET.FIVE                               VOL001 OWN000   31/03/2002  1
DATA.SET.ONE                                VOL000 OWN000   19/02/2002  1
DATA.SET.SEVEN                              VOL002 OWN000   11/03/2002  1
EDG3012I 3          ENTRIES LISTED
```

Figure 89. 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 90:

```
Data set name                                 Volume Owner    Create date Seq
--------------------------------------------- ------ -------- ---------- ----
RMML01.SYSRES.BACKUP                          999000 WOODY    26/02/2002  1
RMML01.SYSRES.BACKUP                          999001 WOODY    26/02/2002  1
RMML01.SYSRES.BACKUP                          999002 WOODY    26/02/2002  1
RMML01.SYSRES.BACKUP                          999003 WOODY    26/02/2002  1
RMML01.SYSRES.BACKUP                          999004 WOODY    26/02/2002  1
EDG3012I 5          ENTRIES LISTED
```

Figure 90. Sample SEARCHDATASET Output

DFSMSrmm also creates a CLIST data set containing the records shown in Figure 91:

```
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 91. Sample CLIST Data Set

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect 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

Before you begin: To use the RMM SEARCHPRODUCT subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

Use the SEARCHPRODUCT subcommand to create a list of software products defined to DFSMSrmm.

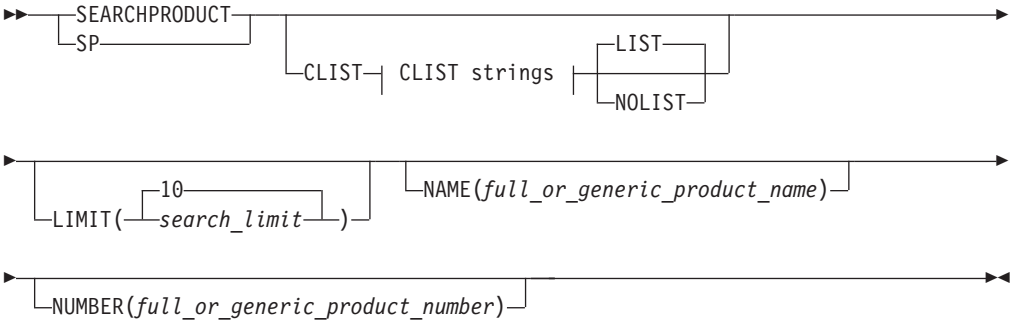
You can restrict how many software products DFSMSrmm displays by specifying the LIMIT operand. DFSMSrmm searches until it reaches your limit or until it lists all software products that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten products.

Table 35 shows the information DFSMSrmm returns for each software product in the list, in the order it is displayed:

Table 35. 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

Syntax



CLIST strings:

Operands

CLIST(*prefix_string,suffix_string***)**
Specifies a CLIST to create a data set of executable commands. You can edit the data set to remove any products you do not want in the list. Then you can run the CLIST at your convenience.

DFSMSrmm returns the software product and version for each record if you do not specify (*prefix_string* and *suffix_string*).

You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). These text strings cannot exceed 255 characters. Separate the *prefix_string* and *suffix_string* using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns.

See “Creating CLISTs of Executable Subcommands” on page 128 for information about the data set used for the CLIST output.

LIMIT(*search_limit***|*)**
Specifies 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 quotation marks 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 quotation marks 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.

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 92.

```
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 92. Sample SEARCHPRODUCT Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHRACK: Creating a List of Shelf Locations

Before you begin: To use the RMM SEARCHRACK subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

Use the SEARCHRACK subcommand to create a list of shelf locations defined in the removable media library. Shelf locations in the removable media library are called rack numbers.

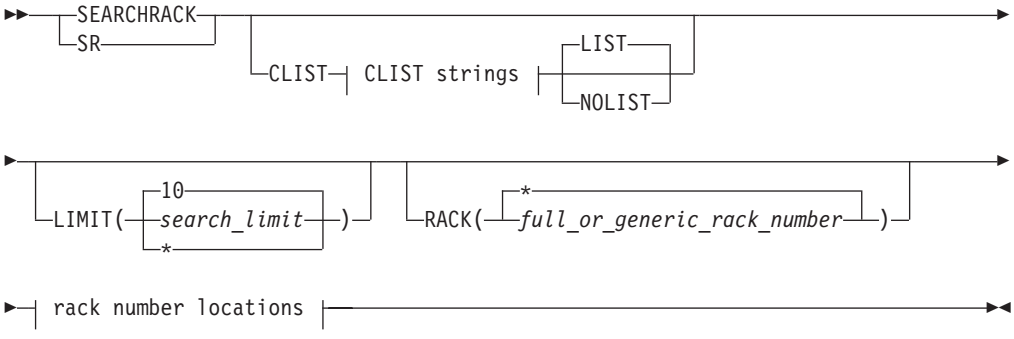
You can restrict the number of rack numbers DFSMSrmm lists by specifying the LIMIT operand. DFSMSrmm searches until it reaches your limit or until it lists all shelf locations that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten shelf locations.

Table 36 shows the information DFSMSrmm returns for each rack number in the list, in the order it is displayed:

Table 36. Information Returned by SEARCHRACK

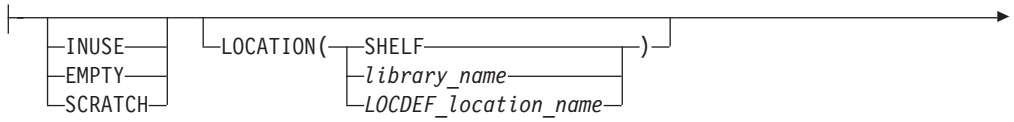
Table Field Name	Description
Rack	Rack number
Medianame	Type of volume
Volume	Volume serial number
Status	Status of the shelf location (one of EMPTY, INUSE, or SCRATCH)
Location	Location where the volume resides

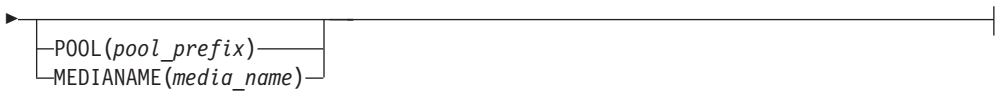
Syntax



CLIST strings:

rack number locations:





Operands

CLIST(*prefix_string,suffix_string***)**
Specifies a CLIST to create a data set of executable commands. You can edit the data set to remove any racks you do not want in the list. Then you can run the CLIST at your convenience.

DFSMSrmm returns the number of the shelf location for each record if you do not specify (*prefix_string* and *suffix_string*). When the rack number contains special characters the value is returned within quotation marks.

You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). These text strings cannot exceed 255 characters. Separate the *prefix_string* and *suffix_string* using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns.

See “Creating CLISTs of Executable Subcommands” on page 128 for information about the data set used for the CLIST output.

EMPTY
Specifies a list of only empty rack numbers in the removable media library. An empty rack number does not contain a volume and is available for use.

INUSE
Specifies a list of only those rack numbers in the removable media library that are in use.

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***|LOCDEF_location)**
Specifies a list that is limited 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.

LOCATION
Specifies a storage location that is defined as a home location.

MEDIANAME(*media_name***)**
Specifies a list that is limited 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 287 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 alphanumeric, national, or special characters followed by an asterisk. A pool ID must be defined by your installation. Enclose it in single quotation marks if it contains any special characters.

Specify POOL(*) to limit the search to rack numbers in the default scratch pool that is defined by your installation for your system. If you do not specify a pool ID, DFSMSrmm lists all rack numbers that match 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 one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. A generic rack number is one-to-five alphanumeric, national, or special characters followed by an asterisk. If you want DFSMSrmm to search through all the rack numbers that are defined to DFSMSrmm, use an asterisk as the rack number.

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
Specifies to list only rack numbers associated with scratch volumes.

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 93 on page 328.

```
Rack/Bin          Media             Current  Moving-in  Moving-out  Old
number   Location  name      Status  volume   volume     volume      volume
-------- --------  --------  ------- -------  ---------  ----------  ------
RAC000   SHELF     3480      SCRATCH  SCR000
RAC001   SHELF     3480      SCRATCH  SCR001
RAC002   SHELF     3480      SCRATCH  SCR002
RAC003   SHELF     3480      SCRATCH  SCR003
RAC004   SHELF     3480      SCRATCH  SCR004
RAC005   SHELF     3480      SCRATCH  SCR005
RAC006   SHELF     3480      SCRATCH  SCR006
RAC007   SHELF     3480      SCRATCH  SCR007
RAC008   SHELF     3480      SCRATCH  SCR008
RAC009   SHELF     3480      SCRATCH  SCR009
RAC010   SHELF     3480      SCRATCH  SCR010
RAC011   SHELF     3480      SCRATCH  SCR011
RAC012   SHELF     3480      SCRATCH  SCR012
RAC013   SHELF     3480      SCRATCH  SCR013
RAC014   SHELF     3480      SCRATCH  SCR014
EDG3203I SEARCH COMPLETE - MORE ENTRIES MAY EXIST
EDG3012I 15         ENTRIES LISTED
```

Figure 93. Sample SEARCHRACK Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 for DFSMSrmm reason codes.

0	Subcommand completed normally.
4	Warning. Subcommand completed but some operands could have been ignored or modified. DFSMSrmmsets 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHVOLUME: Creating a List of Volumes

Before you begin: To use the RMM SEARCHVOLUME subcommand:

- You need READ access to the STGADMIN.EDG.MASTER resource profile.
- In addition, if COMMANDAUTH(DSN) is in effect, you need READ access to the first file data set name in the DATASET class. If there is no first file defined to DFSMSrmm and the volume is in master status or user status, you need READ access to the volume in the TAPEVOL class.

Related Reading: See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

Use the SEARCHVOLUME subcommand 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 37 shows the information DFSMSrmm returns for each volume in the list, in the order it is displayed:

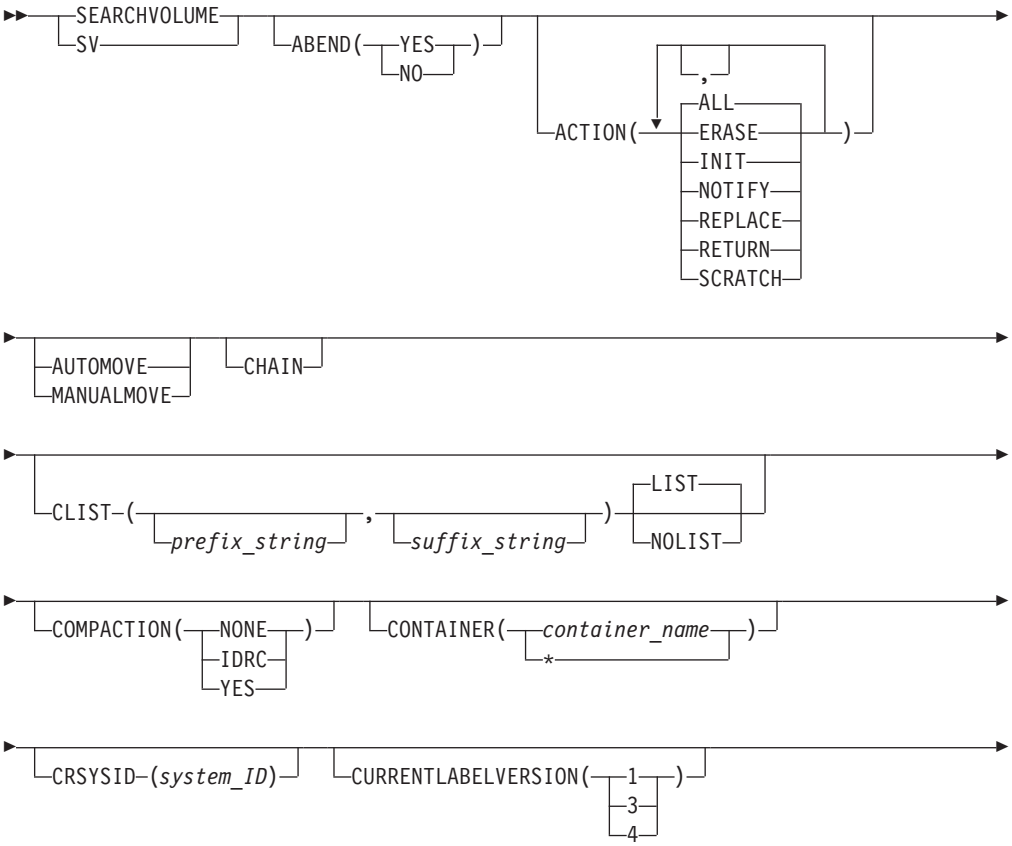
Table 37. Information Returned by the RMM SEARCHVOLUME Subcommand

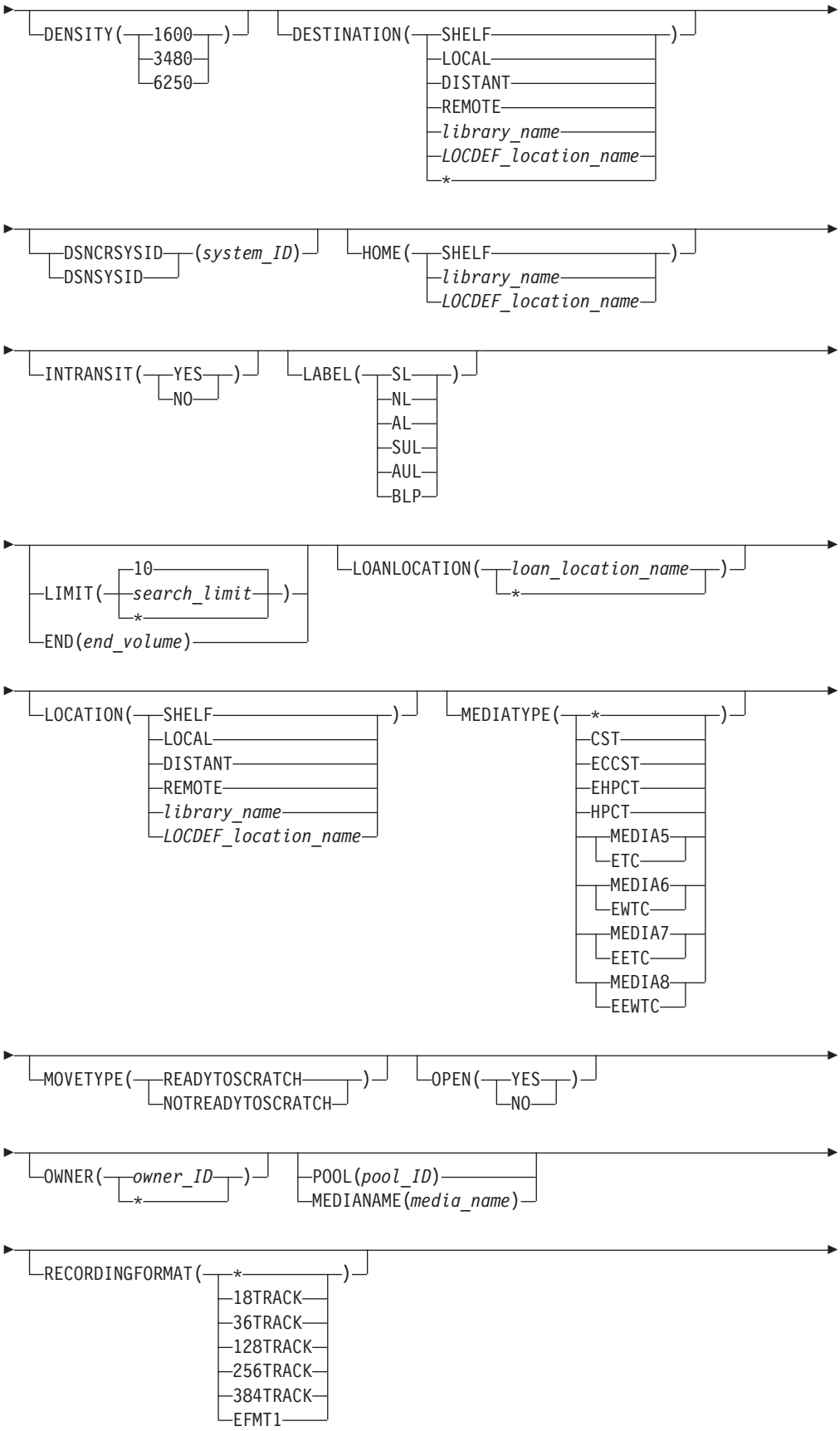
Table Field Name	Description
Volume	Volume serial number
Owner	Owner ID of the volume owner
Rack	Rack number
Assigned date	• Date the volume is assigned to a user • Date the volume is in MASTER status when a non-specific volume is requested in a batch job • Date the volume is returned to scratch status
Expiration date	Date the volume is to be considered for release
Location	Location where the volume resides (one of SHELF, DISTANT, LOCAL, REMOTE, a *library_name*, or a *LOCDEF_location_name*)
Dsets	Number of data sets on the volume

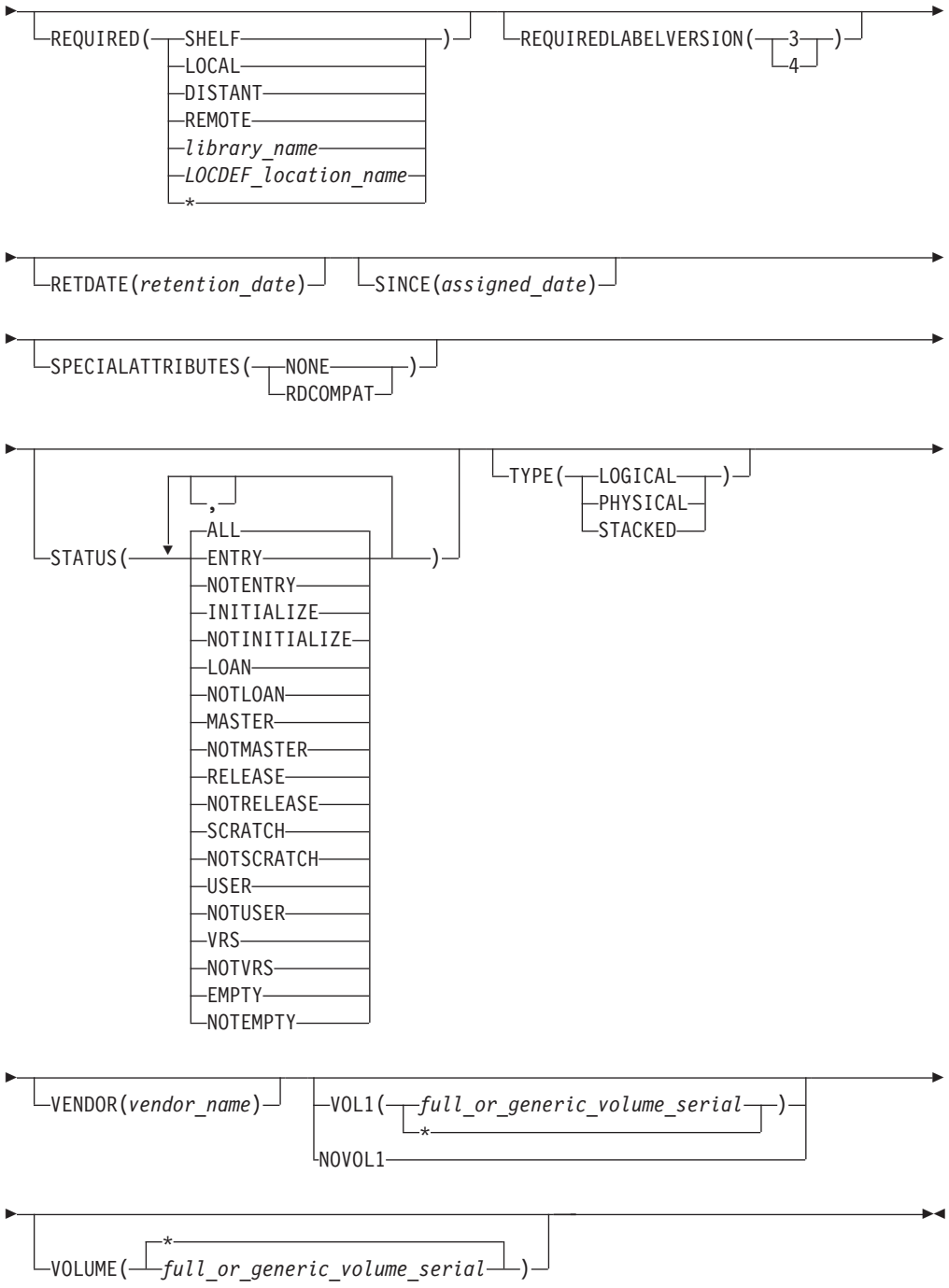
Table 37. Information Returned by the RMM SEARCHVOLUME Subcommand (continued)

Table Field Name	Description
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
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 *library_name*, or a *LOCDEF_location_name*)

Syntax




DENSITY(1600 3480 6250)
DESTINATION(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name *)
DSNCRSYSID DSNSYSID (system_ID)
HOME(SHELF library_name LOCDEF_location_name)
INTRANSIT(YES NO)
LABEL(SL NL AL SUL AUL BLP)
LIMIT(10 search_limit *)
END(end_volume)
LOANLOCATION(loan_location_name *)
LOCATION(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name)
MEDIATYPE(* CST ECCST EHPCT HPCT MEDIA5 ETC MEDIA6 EWTC MEDIA7 EETC MEDIA8 EEWTC)
MOVETYPE(READYTOSCRATCH NOTREADYTOSCRATCH)
OPEN(YES NO)
OWNER(owner_ID *)
POOL(pool_ID)
MEDIANAME(media_name)
RECORDINGFORMAT(* 18TRACK 36TRACK 128TRACK 256TRACK 384TRACK EFMT1)



Operands

ABEND(YES|NO)
Specifies 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.

NOTIFY To list only volumes for which owners must be notified.

SCRATCH To list all volumes to be returned to scratch status.

AUTOMOVE
Specifies a list of volumes that have a move mode of AUTOMOVE which indicates that DFSMSrmm automatic movement processing is in effect.

CHAIN
Specifies 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 thecontrol 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***)**
Specifies a CLIST to create a data set of executable commands. You can edit the data set to remove any volumes you do not want in the list. Then you can run the CLIST at your convenience.

DFSMSrmm returns the volume serial number for each record if you do not specify (*prefix_string* and *suffix_string*). When the volume serial number contains special characters the value is returned within quotation marks.

You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). These text strings cannot exceed 255 characters. Separate the *prefix_string* and *suffix_string* using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns.

See “Creating CLISTs of Executable Subcommands” on page 128 for information about the data set used for the CLIST output.

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***)**
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.

CRSYSID|SYSID(*system_ID***)**
Specifies to list volumes that were created on the named *system_ID*. Specify a system name that is one-to-eight characters long.

CURRENTLABELVERSION(1|3|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***)**
Specifies a list of 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.

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

DSNCRSYSID|DSNSYSID(*system_ID***)**
Specifies to list volumes where the first file was created on the named system ID.

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*|LOCDEF_location)
Specifies 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 resides when it is not stored in a storage location. HOME can be a shelf location in a system-managed tape library or location SHELF. Home location can be a storage location that is defined as a home location.

A home location name is one-to-eight alphanumeric characters starting with a non-numeric character.

INTRANSIT(YES|NO)
Specifies 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)
Specifies 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

LIMIT(*search_limit*|*)
Specifies the number of entries that 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 quotation marks 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	To list
SHELF	Volumes stored in shelf locations in a non-system-managed library.
library_name	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. You cannot specify a distributed library name.
LOCDEF_location_name	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.
LOCAL, DISTANT, or REMOTE	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
Specifies a list of volumes that have a move mode of MANUALMOVE.

MEDIANAME(*media_name***)**
Specifies a list limited 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.

Use the LISTCONTROL subcommand to display media names defined for your location. See “LISTCONTROL: Displaying Parmlib Options and Control Information” on page 287 for more information.

MEDIATYPE(*|CST|ECCST|EHPCT|HPCT|MEDIA5|MEDIA6|MEDIA7|MEDIA8)
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

MEDIA5/ETC IBM TotalStorage Enterprise Tape Cartridge

MEDIA6/EWTC
IBM TotalStorage Enterprise WORM Tape Cartridge 3592

MEDIA7/EETC
IBM TotalStorage Enterprise Economy Tape Cartridge 3592

MEDIA8/EEWTC
IBM TotalStorage Enterprise Economy WORM Tape Cartridge 3592

MOVETYPE(READYTOSCRATCH|NOTREADYTOSCRATCH)
Specifies a list of volumes. 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.

NOVOL1
Specifies a list of volumes that do not have a VOL1 recorded by DFSMSrmm.

OPEN(YES|NO)
Specifies a list limited to volumes with an open condition. Specify OPEN(YES) for physical or logical volumes to find volumes left open by an application. Specify OPEN(NO) with TYPE(STACKED) to list stacked volumes that can be moved.

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 that DFSMSrmm uses to list volumes. DFSMSrmm only lists volumes that are associated with the pool that you specify. A pool ID is one-to-five alphanumeric, national, or special characters that are followed by an asterisk. Pool IDs must be defined by your installation. Enclose it in single quotation marks if it contains any special characters.

Use the LISTCONTROL subcommand with the VLPOOL operand to see the pool IDs that are defined for your installation.

RECORDINGFORMAT(* | 18TRACK | 36TRACK | 128TRACK | 256TRACK | 384TRACK | EFMT1)
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.

18TRACK
Data has been written to the volume in 18 track format. Recording format 18TRACK can be specified with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

36TRACK
Data has been written to the volume in 36 track format. Recording format 36TRACK can be specified with MEDIATYPE(CST) and MEDIATYPE(ECCST) only.

128TRACK
Data has been written to the volume in 128 track format. Recording format 128TRACK can be specified with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

256TRACK
Data has been written to the volume in 256 track format. Recording format 256TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only. Use the 256TRACK operand to limit your search to volumes that are 3590 16-track bi-directional recording with 16 passes.

384TRACK
Data has been written to the volume in 384-track format. A recording format of 384TRACK is valid with MEDIATYPE(EHPCT) and MEDIATYPE(HPCT) only.

EFMT1
Data has been written to the volume in Enterprise Format 1 recording technology. You can only specify EFMT1 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), and MEDIATYPE(MEDIA8).

REQUIRED(*|SHELF| LOCAL| DISTANT| REMOTE|
***library_name*|*LOCDEF_location_name*)**
Specifies a list of volumes that are required to be moved to a specific destination. If you specify REQUIRED(*), and you do not specify the LOCATION operand, DFSMSrmm lists all volumes that currently have a required location. DFSMSrmm does not check if the volume is already in the required location or is already moving to the required location. Use the LOCATION operand to further restrict the list of returned volumes.

Use the following values to identify the volume movement you want DFSMSrmm to display.

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 EDGRMMxx parmlib LOCDEF command. You can enter any location name you want because DFSMSrmm does no checking of the location name.

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 non-system-managed libraries.

REQUIREDLABELVERSION(0|3|4)
Specifies a list limited to volumes that contain the requested required ISO/ANSI label version to be used in the VOL1 label for the volume. Specify 0 for volumes with no required label version.

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, PERMANENT, or WHILECATLG. 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. To obtain a list of volumes that have a permanent expiration date and that are not retained by vital record specifications, specify the expiration dates 1999/365 or 1999/366.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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 58 for information about how DFSMSrmm calculates retention dates.

SPECIALATTRIBUTES(NONE|RDCOMPAT)
Specifies a list limited to volumes with special attributes associated with the tape volume.

RDCOMPAT To list only those volumes with the RDCOMPAT special attribute

NONE To list only those volumes with no special attributes

SINCE(*assigned_date***)**
Specifies a list limited to volumes assigned to a user or volumes returned to scratch status after the *assigned_date*.

Supply the year and day in one of two forms. We recommend that you use the yyyy/ddd format rather than the yyddd format for dates.

- *yyyy/ddd*, where *yyyy* is the four-digit number for the year. The maximum allowable value for *yyyy* is 9799. *ddd* is the three-digit number for the day of the year, such as 2004/001. The slash is required.
- *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 04001. When you use the *yyddd* format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

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, EMPTY, ENTRY, INITIALIZE, MASTER, RELEASE, SCRATCH, USER, VRS, NOTEMPTY, 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.

EMPTY To list volumes that are empty. A stacked volume is empty when it contains no volumes. Any other type of volume is empty when it contains no data sets.

ENTRY To list volumes already defined to DFSMSrmm before being entered into an automated tape library 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 into an automated tape library 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.

NOTEMPTY To exclude stacked volumes that contain no volumes or other volumes that contain no data sets.

NOTENTRY Specifies a list that excludes volumes already defined to DFSMSrmm. These volumes are ones that have not yet been entered into an automated tape library for use as scratch volumes.

NOTINITIALIZE
Specify to exclude volumes that must be initialized before they are available for use as scratch volumes.

NOTMASTER Specify to exclude volumes that are currently in master status.

NOTLOAN Specify to exclude volumes that are currently stored in loan locations.

NOTRELEASE
Specify to exclude volumes that are pending release and might require a release action before they can be released.

NOTSCRATCH
Specify to exclude volumes that are currently in scratch status.

NOTUSER Specify to exclude volumes from the list returned by the SEARCHVOLUME request that are currently in user status.

NOTVRS Specify to exclude volumes that are retained by vital record specifications.

TYPE(LOGICAL|PHYSICAL|STACKED)
Specifies a list that limits the search to volume type. 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.

VENDOR(*full_or_generic_vendor_name***|*)**
Specifies a list of volumes that have a vendor name provided. A full vendor name is one-to-eight alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. A generic vendor name is one-to-seven alphanumeric, national, or special characters followed by an asterisk. Specify VENDOR(*) to obtain a list of all volumes with any provided vendor name.

There is no default.

VOLUME(*full_or_generic_volume_serial***|*)**
Specifies the serial number of the volume that is being searched. A full volume serial number is one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. A generic volume serial is one-to-five alphanumeric, national, or special characters followed by an asterisk. Specify an asterisk to search all volumes that match the specified search criteria.

VOL1(*full_or_generic_volume_serial***|*)**
Specifies a list of volumes that have a VOL1 recorded by DFSMSrmm. A full volume serial number is one-to-six alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters. A generic volume serial is one-to-five alphanumeric, national, or special characters followed by an asterisk. Specify VOL1(*) to obtain a list of all volumes with any VOL1 values.

Examples

Task: Create a list of all duplicate volumes.

Command:

```
RMM SEARCHVOLUME VOL1(*)
```

Output: DFSMSrmm displays a list such as the one shown in Figure 94. To obtain the VOL1 value for each of the volumes in the output list, issue an RMM LISTVOLUME command for each volume.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
R00001 D044412  R00001 2002/182   2002/187   SHELF    0     M
R00002 D044412  R00002 2002/182   2002/187   SHELF    0     M
R00003 D044412  R00003 2002/182   2002/187   SHELF    0     M
EDG3012I 3          ENTRIES LISTED
```

Figure 94. Sample SEARCHVOLUME Output Listing Duplicate Volumes

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 the one shown in Figure 95.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                      date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
VOL600 WEISSENB RAC500 06/11/2004 11/11/2002 SHELF    0     UR SI
EDG3011I 1 ENTRY LISTED
```

Figure 95. Sample SEARCHVOLUME Output Listing Owned Volumes

Task: Search for all volumes moving from an automated tape library 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 96.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                      date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
9990A5 ZWT01    9990A5 15/01/2003 20/01/2003 ATL      1     M        SHELF
9990A6 ZWT01    9990A6 15/01/2003 20/01/2003 ATL      1     M        SHELF
EDG3012I 2          ENTRIES LISTED
```

Figure 96. Sample SEARCHVOLUME Output Listing Volumes to Eject

DFSMSrmm also creates a CLIST data set containing the records shown in Figure 97.

```
RMM CHANGEVOLUME 9990A5 EJECT(BULK)
RMM CHANGEVOLUME 9990A6 EJECT(BULK)
```

Figure 97. 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(2002/300)
```

Output: DFSMSrmm displays a list such as the one shown in Figure 98 on page 343.

```
 Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                        date       date
 ------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
 KIM002 KIMBERLY ABC002 06/04/1993 10/01/2002 SHELF    1     M
 KIM006 KIMBERLY DAMW08 06/04/1993 10/01/2002 SHELF    1     M
 111001 LESLEY01 D65B35 06/18/1994 10/01/2002 SHELF    1     M        REMOTE
 111013 LESLEY01 D65B36 06/18/1994 10/01/2002 SHELF    1     M        REMOTE
```

Figure 98. 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(ATL1)-
      TYPE(LOGICAL) CLIST
```

Output: DFSMSrmm displays sample SEARCHVOLUME output in Figure 99. DFSMSrmm displays sample CLIST output in Figure 100.

```
A00099 LESLEY01 07/30/1999 08/26/1999 SHELF 1 M VTS1
```

Figure 99. Sample SEARCHVOLUME Output Building a List of Logical Volumes

```
CONT01,A00099
```

Figure 100. Sample SEARCHVOLUME CLIST Output for a List of Logical Volumes

Task: Ensure that existing logical volumes are identified to DFSMSrmm as logical volumes.

Command:

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) LOCATION(ATL1)-
      CLIST('RMM CHANGEVOLUME ',' TYPE(LOGICAL)')
EXEC EXEC.RMM.CLIST
```

Output: DFSMSrmm displays CLIST output as shown in Figure 101.

```
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 101. Sample SEARCHVOLUME CLIST 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 CLIST output as shown in Figure 102.

```
RMM CHANGEVOLUME RFA050 LOCATION(LIB2)
RMM CHANGEVOLUME RFA051 LOCATION(LIB2)
RMM CHANGEVOLUME RFA052 LOCATION(LIB2)
RMM CHANGEVOLUME RFA053 LOCATION(LIB2)
RMM CHANGEVOLUME RFA054 LOCATION(LIB2)
RMM CHANGEVOLUME RFA055 LOCATION(LIB2)
RMM CHANGEVOLUME RFA056 LOCATION(LIB2)
RMM CHANGEVOLUME RFA057 LOCATION(LIB2)
RMM CHANGEVOLUME RFA058 LOCATION(LIB2)
RMM CHANGEVOLUME RFA059 LOCATION(LIB2)
```

Figure 102. 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 ',' CMOVE') OWNER(*) LIMIT(*)
```

Output: DFSMSrmm displays CLIST output as shown in Figure 103.

```
RMM CHANGEVOLUME RFA199 CMOVE
```

Figure 103. Confirming Volume Moves from One Library to Another Library

Task: List any volumes in the volume serial number range from XWW126 through XXW131 inclusive.

Command:

```
RMM SEARCHVOLUME VOLUME(XWW126) END(XXW131) OWNER(*)
```

Output: DFSMSrmm displays a list such as the one in Figure 104.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
XWW126 TRISH           1999/307   1999/312   SHELF    0     M
XWW127 TRISH           1999/307   1999/312   SHELF    0     M
XXW130 TRISH           1999/307   1999/312   SHELF    0     U
XXW131 TRISH           1999/307   1999/312   SHELF    0     U
EDG3012I 4          ENTRIES LISTED
```

Figure 104. Listing Volumes within a Range of Volume Serial Numbers

Task: List any volumes in the volume serial number range using a generic volume serial number.

Command:

```
RMM SEARCHVOLUME VOLUME(XWW*) END(XXW) OWNER(*)
```

Output: DFSMSrmm displays a list such as the one in Figure 105.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
XWW125 TRISH           1999/307   1999/312   SHELF    0     M
XWW126 TRISH           1999/307   1999/312   SHELF    0     M
XWW127 TRISH           1999/307   1999/312   SHELF    0     M
EDG3012I 3          ENTRIES LISTED
```

Figure 105. Using a Generic Volume Serial Number to Obtain a List of Volumes within a Range of Volume Serial Numbers

Task: List any volumes that belong to the owner TRISH but stop with volume serial number 123456.

Command:

```
RMM SEARCHVOLUME OWNER(TRISH) END(123456)
```

Output: DFSMSrmm displays a list such as the one in Figure 106.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
XWW125 TRISH           1999/307   1999/312   SHELF    0     M
XWW126 TRISH           1999/307   1999/312   SHELF    0     M
XWW127 TRISH           1999/307   1999/312   SHELF    0     M
XXW130 TRISH           1999/307   1999/312   SHELF    0     U
XXW131 TRISH           1999/307   1999/312   SHELF    0     U
XXW132 TRISH           1999/307   1999/312   SHELF    0     U
123456 TRISH           1999/307   1999/312   SHELF    0     U
EDG3012I 7          ENTRIES LISTED
```

Figure 106. Listing Volumes for an Owner within a Range of Volume Serial Numbers

Task: Create a volume export list.

Command:

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) LOCATION(ATLBA035) REQUIRED(MAZ) -
     OWNER(*) CLIST('',',MAZ') INTRANSIT(N)
```

Output: Figure 107 shows the output from the RMM SEARCHVOLUME subcommand request. Figure 108 on page 346 shows the resulting CLIST information.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
RFA055 MTHUM           02/24/2003 03/01/2003 ATLBA035 1     UV
RFA056 MTHUM           02/24/2003 03/01/2003 ATLBA035 1     UV
2          ENTRIES LISTED
```

Figure 107. Listing Volumes for the Export List

```
RFA055,MAZ
RFA056,MAZ
```

Figure 108. CLIST Information for the Export List

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

SEARCHVRS: Creating a List of Vital Record Specifications

Before you begin: To use the RMM SEARCHVRS subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

Use the SEARCHVRS subcommand to create a list of vital record specifications.

Table 38. 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 that is 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.

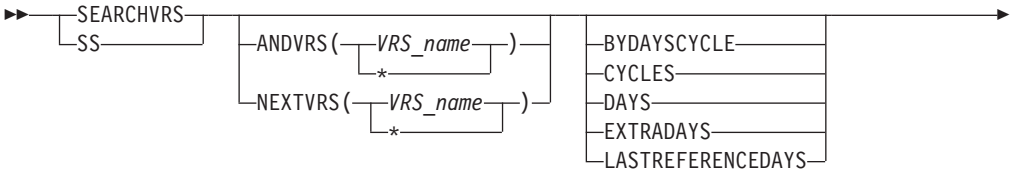
Use the LIMIT operand to restrict how many vital record specifications DFSMSrmm lists. DFSMSrmm searches for vital record specifications until it reaches the limit you specify or until it lists all vital record specifications that match your search criteria. DFSMSrmm lists a maximum of ten vital record specifications if you do not specify the LIMIT operand.

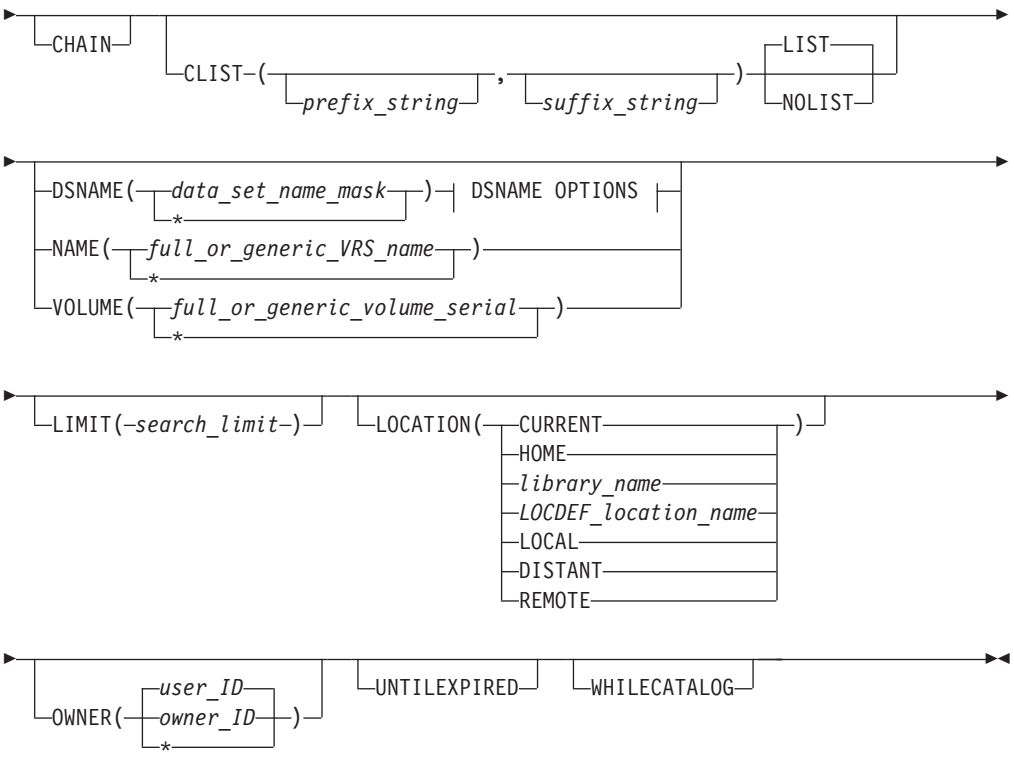
Table 39 shows the information DFSMSrmm returns for each vital record specification in the list, in the order it is displayed:

Table 39. Information Returned by SEARCHVRS

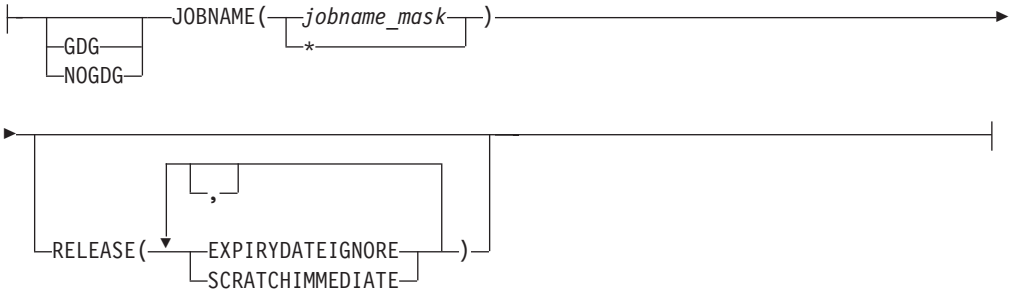
Table Field Name	Description
Vital Record Specification	Volume serial number, data set name mask, or vital record specification name
Job name	Name of the job that created the data set
Type	Type of vital record specification (one of DSN, VOL or NAME)
Owner	Owner ID
Location	Location where the volume retained by the vital record specification resides
Next VRS	Name of next name vital record specification to which this vital record specification is linked

Syntax





DSNAME OPTIONS:



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***)**
Specifies a CLIST to create a data set of executable commands. You can edit the data set to remove any vital record specifications you do not want in the list. Then you can run the CLIST at your convenience.

DFSMSrmm returns the data set name, volume serial number, or vital record specification name for each record if you do not specify (*prefix_string* and *suffix_string*). Specify the DSNAME, NAME, or VOLUME operands to limit the list to only data set, name, or volume vital record specifications.

You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying (*prefix_string* and *suffix_string*). These text strings cannot exceed 255 characters. Separate the *prefix_string* and *suffix_string* using a blank or a comma between the text strings. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns.

See “Creating CLISTs of Executable Subcommands” on page 128 for information about the data set used for the CLIST output.

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.

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***|*)**
Specifies a job name that 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***)**
Specifies to list only those vital record specifications with the location indicated. *library_name* is any eight-character name meeting the system-managed library 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***l*)**
Specifies a volume serial number. A full volume serial number is one-to-six alphanumeric, national, 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. Enclose a full or generic volume serial number in single quotation marks if it contains any special characters.

VOLUME cannot be used 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: 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 109.

```
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 109. Sample SEARCHVRS Output

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 110.

```
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 110. Sample SEARCHVRS Output

Task: Create a list of vital record specifications where the NEXTVRS value is VRS002.

Command:

```
RMM SEARCHVRS NAME(*) OWNER(*) NEXT(VRS002)
```

Output: DFSMSrmm displays a list such as Figure 111.

```
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 111. Sample SEARCHVRS Output

Return Codes

See Chapter 10, “DFSMSrmm Return Codes and Reason Codes,” on page 355 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. The data is incomplete or incorrect and the TSO user has set NOPROMPT.
24	The TSO subcommand is not APF authorized.
28	The user pressed the attention key.

Chapter 10. DFSMSrmm Return Codes 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 40 lists the return codes issued by DFSMSrmm.

Table 40. 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 112, 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 112. Testing RMM TSO LISTVOLUME Subcommand Example 1

You can also code REXX statements, as shown in Figure 113, 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 113. Testing RMM TSO LISTVOLUME Subcommand Example 2

© Copyright IBM Corp. 1992, 2004

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, or returns messages, for return code 20 only when DFSMSrmm processing determines an error other than value range or naming restrictions. In other cases, it is the TSO parse function that provides the information messages for parse errors. See Table 23 on page 163 for command abbreviations that are also used in Table 41.

Table 41 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 *z/OS MVS System Messages, Vol 5 (EDG-GFS)*.

Table 41. 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	SB SD SP SR SS SV	Search processing indicates more records might exist
04	04		SB SD SP SR SS SV	Search processing indicates fewer records returned than were 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	SB 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		SB SD SP SR SS SV	Insufficient storage for search processing. More records might exist
04	12	EDG3328	SD SV	A record in a chain is not found.
04	158	EDG3304	CV	Move is not pending.
04	160	EDG3305 EDG3306 EDG3307 EDG3285	CV	Move is not confirmed.
04	162	EDG3286	CV	Action is not confirmed.
04	164	EDG3287	CV	Action is not pending.
12	00		All	I/O error on DFSMSrmm control data set

Table 41. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
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 EDG3018	AB AR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the COUNT operand is greater than 1 and the command fails.
12	18	EDG3200	AO	Owner already exists
12	18	EDG3200	AP	Software product already exists
12	18	EDG3200	AB 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 DD LD	Data set does not exist
12	18	EDG3201	CO DO LO	Owner does not exist
12	18	EDG3201	CP DP LP	Software product does not exist
12	18	EDG3201	CV DV LV	Volume does not exist
12	18	EDG3201	DB DR LB LR	Rack or bin number does not exist
12	18	EDG3201	DS LS SS	Vital record specification does not exist
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	Backup in progress - DFSMSrmm control data set cannot be changed
12	26	EDG3009	AP AS AV CP CV GV SD SV	Owner does not exist
12	28	EDG3223	AD SD SV	Volume does not exist
12	30	EDG3224	AD CD CV 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	SB SR	Incorrect rack number key in data area
12	36	EDG3228	SB SR	Incorrect bin number key in data area
12	38	EDG3017	AB AR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the COUNT operand is greater than 1 and the command fails.

Table 41. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	38	EDG3229	AV CV DB DR	Rack number does not exist or is not empty
12	40	EDG3230	AV CV GV SB SR 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 CV	Pool or unit mismatch
12	56	EDG3238	AV CV	Pool is full
12	58	EDG3239	AB AR DB DR LB LR SB SR	Incorrect storage location ID in data area
12	60	EDG3240	AV 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 CV	Both Pool and rack specified in the data area
12	68	EDG3017 EDG3018	AB AR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the COUNT operand is greater than 1 and the command fails.
12	68	EDG3019 EDG3020	DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the COUNT operand is greater than 1 and the command fails.
12	68	EDG3244	AV	Count too big. Volume serial number generated > 999999
12	68	EDG3244	AB AR DB DR	Count too big. Rack number or bin number generated > 999999
12	70	EDG3017 EDG3018	AB AR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the COUNT operand is greater than 1 and the command fails.
12	70	EDG3019 EDG3020	DB DR	DFSMSrmm issues values for EDG@CNT and EDG@RCK when the COUNT operand is greater than 1 and the command fails.
12	70	EDG3245	AB AR AV DB DR	Count too big. Volume serial numeric suffix or rack number suffix exhausted
12	72	EDG3246	AV CV	Previous volume already has a next volume
12	74	EDG3247	DB DR	Bin not empty
12	76	EDG3248	DV	Volume is already a scratch volume
12	78	EDG3249	AV 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 CV	Unknown previous volume

Table 41. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	88	EDG3253	AV 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	SB SD SP SR SS SV	Unable to open CLIST data set
12	118	EDG3269	SB SD SP SR SS SV	CLIST organization not sequential or partitioned
12	120	EDG3270	SB SD SP SR SS SV	CLIST data set too small
12	122	EDG3277	AV CV	Manual cartridge entry failed
12	124	EDG3278	AV 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 DV	Volume eject 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 CV	Rack number and volser are not the same
12	140	EDG3292	AV 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.

Table 41. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
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	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 CD 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 DB	Bin number does not exist or is not empty.
12	180	EDG3266	AS	COUNT is too small.
12	182	EDG3267	AS	COUNT must equal STORENUMBER.
12	184	EDG3297	AS	STORENUMBER is missing.
12	186	EDG3325	SD SV	CHAIN specified without a 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 CV	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
12	202	EDG3334	AV CV DV	Library type cannot be determined. The library must be correctly configured with at least one tape drive operational.
12	204	EDG3335	CV	New volume is not supported for the volume type
12	206	EDG3336	DV	Stacked volume is not empty
12	208	EDG3337	AV CV	Stacked volume container does not exist and stacked volume support is enabled
12	210	EDG3338	AV CV	Initialize action is not supported for a stacked volume.
12	212	EDG3339	DV	RELEASE option is not supported for a stacked volume
12	214	EDG3340	AV CV	Storage group and library combination are not valid
12	216	EDG3341	AV CV	System-managed library information is not consistent with the type of volume
12	218	EDG3342	CV	A virtual export cannot be performed because the export token does not match or the volume was never in the container and the stacked volume is export recorded.

Table 41. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
12	220	EDG3205	AB AD AO AP AR AS AV CD CO CP CV DB DD DO DP DR DS DV GV	Journal is locked.
12	222	EDG3343	AV	Status SCRATCH is not supported for a stacked volume.
12	224	EDG3344	AV	Volume catalog (TCDB) information conflicts with Library Manager information.
12	226	EDG3345	AV CV	Media type is required for manual cartridge entry.
12	228	EDG3346	CV DR	Subcommand failed because extended bin support is enabled.
12	230	EDG3347	CV	Cancel move failed because bin is already moving.
12	232	EDG3351	CV	Start move rejected, another volume is already moving out of bin.
12	234	EDG3352	CV	Bin support is not supported for a logical volume or a volume in a container.
12	236	EDG3353	CV	Unexpected return code xx and reason Code yyyy from subsystem request.
12	238	EDG3354	CV	Container change rejected as the volume is moving.
12	240	EDG3355	AV CV	VOL1 is not supported for NL tapes, logical volumes, or stacked volumes
12	244	EDG3357I	CV	WWID cannot be changed once it is set.
20	0		All	Parse error for which we have no more specific reason.
20	2	EDG3106	AD AV CD CV GV LC	SECLEVEL value is not defined to DFSMSrmm
20	4	EDG3107	AD AV CD CV GV LC	Do not use SECLEVEL because your installation has no security classes defined.
20	6	EDG3272	AV CV GV AR AS	LOCATION operand specified a library that is not defined to DFSMS.
20	8	EDG3274	AS CV GV	LOCATION operand specified a library and libraries are not supported.
20	10	EDG3273	AV CV	STORAGEGROUP operand value is not defined to DFSMS.
20	12	EDG3275	AV CV	STORAGEGROUP operand is not supported.
20	14		AS DS LS SD SS	DSNAME value does not meet DFSMSrmm generic naming restrictions.
20	16	EDG3272	CV	HOME operand specified an allocation name that is not defined to DFSMS, or a storage location that is not allowed as a home location.
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 AR	Media name is not valid for the specified location.

Table 41. DFSMSrmm Reason Codes (continued)

Return Code	Reason Code	Message Number	Issuing Command	Description
20	24		AS DS LS SS	JOBNAME value does not adhere to DFSMSrmm generic naming rules.
20	26		AS CD SS	JOBNAME value does not adhere to DFSMSrmm naming rules.
20	28	EDG3348	AV CV GV	LOCATION operand specified a storage location name that is not defined as a home location.
20	30	EDG3348	CV	HOME operand specified a storage location name that is not defined as a home location.
20	32	EDG3349	AV CV GV	LOCATION operand specified a storage location name, and storage locations are not allowed.
20	34	EDG3349	CV	HOME operand specified a storage location name, and storage locations are not allowed.
20	36	EDG3350	AR	LOCATION is not bin-managed.
20	38		AV, CV, SV	An unsupported media type is specified.
20	40		AV, CV, SV	An unsupported recording technology is specified.

Chapter 11. DFSMSrmm REXX Variables

This chapter contains information you can use to create your own REXX execs or procedures to use with DFSMSrmm.

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 that 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 that are 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 42 lists all the variables in RMM subcommand order that you can use in your REXX execs.

Table 42. 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@DDT EDG@DESC EDG@DSN EDG@LOC EDG@NAME EDG@NVRS EDG@OWN EDG@PRTY EDG@RC EDG@RET EDG@RWC EDG@SC1 EDG@TYP EDG@VDD EDG@VOL EDG@VRC EDG@VSS EDG@UEX EDG@VJBN
CHANGEDATASET	EDG@RC
CHANGEOWNER	EDG@RC
CHANGEPRODUCT	EDG@RC

© Copyright IBM Corp. 1992, 2004

Table 42. 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@MIV EDG@MOV EDG@OVOL EDG@RC EDG@RCK EDG@RST EDG@VOL
LISTCONTROL ACTIONS	EDG@ACT[1] EDG@AST[1] EDG@RC
LISTCONTROL CNTL	EDG@ACS EDG@AUD EDG@BDT EDG@BTM EDG@CDSU EDG@CSHN EDG@CSIP EDG@CSVE EDG@DBN EDG@CDS EDG@CSDT EDG@CSTM EDG@DDT EDG@DRP EDG@DTE EDG@DTM EDG@EBIN 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@JBDT EDG@JBTM EDG@JDS EDG@JRNS EDG@JRNU EDG@LBN EDG@LCT EDG@LRK EDG@MDS EDG@MDT EDG@MRP EDG@MTM EDG@MTP EDG@NOT EDG@OPM EDG@PACS EDG@RBN EDG@RC EDG@RCF EDG@RDT EDG@RTM EDG@RUB EDG@SAT EDG@SDT EDG@SID EDG@SLM EDG@SOSD EDG@SOSP EDG@SOST EDG@SSM EDG@STM EDG@VDT EDG@VTM EDG@XDTJ EDG@XTM EDG@X100 EDG@X200
LISTCONTROL LOCDEF	EDG@LDDF[1] EDG@LDLC[1] EDG@LDLT[1] EDG@LDMN[3] EDG@LDMT[1] EDG@LDPR[1] EDG@RC
LISTCONTROL MNTMSG	EDG@MID[1] EDG@OPL[1] EDG@OVL[1] EDG@RC EDG@SMI[1]
LISTCONTROL MOVES	EDG@MFR[1] EDG@MST[1] EDG@MTO[1]EDG@MTY[1]

Table 42. TSO Subcommand Variables by RMM Subcommand (continued)

Subcommand	Variable Name
LISTCONTROL OPTION	EDG@ACCT EDG@AUD EDG@BLP EDG@BKPP EDG@CATS EDG@CDS EDG@CMDD EDG@CMDO EDG@CRP EDG@DRP EDG@DSPD EDG@DSPM EDG@DTE EDG@IPL EDG@JDS EDG@JRNF EDG@LCT EDG@LCTK EDG@MDS EDG@MEDN EDG@MOP EDG@MRP EDG@MSGF EDG@MVBY EDG@OPM EDG@NOT EDG@PDAC EDG@PDA EDG@PDAC EDG@PDAL EDG@PDAS EDG@PSFXEDG@RC EDG@RCF EDG@RTBY EDG@RUB EDG@SID EDG@SLM EDG@SMP EDG@SMU EDG@SOSP EDG@SRHN EDG@SRIP EDG@SRPN EDG@SRTK EDG@SSM EDG@SSTY EDG@UNC EDG@VACT EDG@VCHG EDG@VMIN EDG@VRJ EDG@VRSL
LISTCONTROL REJECT	EDG@GRK[1] EDG@RC EDG@TAC[1]
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@ACT[1] EDG@MEDN[1] EDG@MOP EDG@PDS[1] EDG@PID[1] EDG@PLN[1] EDG@PRF[1] EDG@PSN[1] EDG@PTP[1] EDG@RC EDG@SCRM[1] EDG@XDC[1]
LISTDATASET	EDG@ABND EDG@BLKC EDG@BLKS EDG@BLKT EDG@CDT EDG@CJBN EDG@CLS EDG@CPGM EDG@CTLG EDG@CTM EDG@DEV EDG@DLR EDG@DLW EDG@DPCT EDG@DSEQ EDG@DSN EDG@FILE EDG@LDD EDG@LDEV EDG@LPGM EDG@LJOB EDG@LRCL EDG@LSTP EDG@MC EDG@NME EDG@OWN EDG@OXD 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@XDT 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]

Table 42. TSO Subcommand Variables by RMM Subcommand (continued)

Subcommand	Variable Name
LISTRACK	EDG@LOC EDG@MEDN EDG@PID EDG@RC EDG@RCK EDG@RST EDG@VOL
LISTVOLUME ACCESS	EDG@ID1-12 EDG@LCID EDG@MVS EDG@OAC EDG@RC EDG@VAC EDG@VM
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@RBYS EDG@RC EDG@RCK EDG@RTDT EDG@SGN EDG@STVC EDG@VCAP EDG@VNDR EDG@VOL EDG@VOLT EDG@VOL1 EDG@VPCT EDG@VRSI EDG@VRXI EDG@VST EDG@WWID EDG@XDT
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 EDG@VWMC
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@TYP EDG@UEX EDG@VANX EDG@VDD EDG@VJBN EDG@VRC EDG@VRS EDG@VRSI EDG@VRXI EDG@VSS
SEARCHBIN	EDG@MEDN[1] EDG@MIV EDG@MOV EDG@OVOL 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@OXD 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]

Table 42. TSO Subcommand Variables by RMM Subcommand (continued)

Subcommand	Variable Name
SEARCHRACK	EDG@MEDN[1] EDG@PID EDG@RC EDG@RCK[1] EDG@RST[1] EDG@VOL[1]
SEARCHVOLUME	EDG@ADT[1] EDG@ADTJ[1] EDG@AVL[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@SYS EDG@TYPF EDG@TYPT [1] EDG@VNDR EDG@VOL[1] EDG@VST[1] EDG@WWID EDG@XDT[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@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 363.

TSO Subcommand Variables by Name

Table 43 lists the variables you can use in your REXX execs by variable name. See Table 23 on page 163 for command abbreviations that are used in this section.

Table 43. 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@ACS	LC OPT	SMSACS option	YES or NO
EDG@ACT	LV	Actions to be performed on release S=SCRATCH R=REPLACE I=INIT E=ERASE N=NOTIFY O=OWNER	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
	LC VLPOOL[1]	Action name	N or blank
EDG@ADT	LV SV[1]	Assigned date	Date format

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@ADTJ [1]	SV	Assigned date	Julian date format
EDG@AD1	LO	Volume owner's address line 1	40 characters
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	8 characters (hh:mm:ss)
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	Number of data set blocks	Numeric: 1 - 999999
EDG@BLKS	LD SD	Size of data set blocks	Numeric: 1 - 99999
EDG@BLKT	LD	Total block count	10 characters
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	8 characters (hh:mm:ss)
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@CDSU	LC CNTL	Control data set percentage used	Numeric: 0 - 100
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@CMDD	LC OPT	Command authorization based on data set name	3 characters: YES or NO

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@CMDO	LC OPT	Command authorization based on owner name	3 characters: YES or NO
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@CPGM	LD	Creating program name	8 characters
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
EDG@CSHN	LC CNTL	Client/server local host name	63 characters: 1 to 63 alphanumeric characters including hyphen, period, and blank
EDG@CSIP	LC CNTL	Client/server local IP address	15 characters: 1 to 15 numeric characters including period and blank
EDG@CSTM	LC CNTL	Catalog synchronize time	8 characters (hh:mm:ss)
EDG@CSVE	LC CNTL	Stacked volume enablement status	8 characters: ENABLED, DISABLED, NONE, MIXED
EDG@CTLG	LD	Catalog status	One of UNKNOWN, YES, or NO
EDG@CTM	LD SD[1]	Data set create time	8 characters (hh:mm:ss)
	LV	Volume create time	8 characters (hh:mm:ss)
EDG@CNTR	LV	Container in which the resource is stored (stacked volume)	6 characters
EDG@DBIN	LV	Destination bin number	Numeric: 0 - 999999 or 6 alphanumeric characters
EDG@DBMN	LV	Destination bin media name	8 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

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
	LS	Vital record specification description	30 characters
EDG@DEST	LV SV[1]	Destination name	One of SHELF, built-in storage location name, installation-defined storage location name, 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@DPCT	LD	Percentage of volume	0 - 100
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 - 65535
EDG@DSEQ	LD	Data set sequence number	Numeric: 0 - 65535
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
EDG@DSTT	LV	Destination type	One of AUTO, MANUAL, STORE, or blank
EDG@DSYS	LV VOL, SV[1]	Creation system ID for first file	1-to-8 characters
EDG@DTE	LC	Installation date format	One of A, E, I, or J
EDG@DTM	LC	Last inventory management run time	8 characters (hh:mm:ss)
EDG@EMN	LO	Owner's node	8 characters
EDG@EBIN	LC CNTL	Extended bin enable status	8 characters: one of ENABLED, DISABLED
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

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
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	Numeric: 0 - 65535
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 12
	CV	OAM reason code	OAM return code l2
	DV	OAM reason code	OAM return code l2
	GV	OAM reason code	OAM return code l2
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
EDG@GRK	LC[1]	Generic rack number	6 characters
EDG@HLOC	LV SV[1]	Home location name	8 characters
EDG@HLOT	LV	Home location type	One of AUTO, MANUAL, or blank
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@JBDT	LC CNTL	Last journal back up date	Date format

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@JBTM	LC CNTL	Last journal back up time	8 characters (hh:mm:ss)
EDG@JDS	LC	Journal name	44 characters
EDG@JOB	LV	Job name	8 characters
EDG@JRNF	LC	JOURNALFULL parmlib operand value	Numeric 0 - 99
EDG@JRNS	LC CNTL	Journal status	8 characters: ENABLED, LOCKED, or DISABLED
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@LCTK	LC OPTION	Local tasks	3 characters: 1 to 3 numeric characters
EDG@LDD	LD	Last used DD name	8 characters
EDG@LDDF	LC[1]	Existence of a location definition	YES or NO
EDG@LDEV	LD, 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 LOCDEF1[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@LJOB	LD	Last used job name	8 characters
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
	LR SR[1]	Location of volume, rack number	SHELF, or an 8 character name of a system-managed library name

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
	AS LS SS[1]	Location name	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@LPGM	LD	Last used program name	8 characters
EDG@LRCL	LD	Data set Logical Record Length (LRECL)	Numeric: 0 - 99999
EDG@LRK	LC	Number of LIBRARY rack numbers	Numeric: 0 - 9999999999
EDG@LSTP	LD	Last used step name	8 characters
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 LB LC[1] LC OPT LR LV SB[1] SR[1] SV[1]	Media name	8 characters
EDG@MEDR	LV SV[1]	Tape recording technology	8 characters. One of *, 18TRACK, 36TRACK, 128TRACK, 256TRACK, 384TRACK, EFMT1
EDG@MEDT	LV SV[1]	Tape media type	One of *, CST, ECCST, EHPCT, HPCT, ETC, EETC, EWTC, and EEWTC
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@MIV	LB, SB[1]	Moving-in volume	6 characters
EDG@MOP	LC, LC VLPOOL[1]	Masteroverwrite	One of ADD, MATCH, LAST, or USER

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@MOV	LB, SB[1]	Moving-out volume	6 characters
EDG@MOVM	LV	Move mode	AUTO or MANUAL
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	8 characters (hh:mm:ss)
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@MVBY	LC	Move by value	6 characters: VOLUME or SET
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@NLOT	LV	Required location type	One of AUTO, MANUAL, STORE, or blank
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	6 characters
EDG@NVRS	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@OLOT	LV	Old location type	One of AUTO, MANUAL, STORE, INCTNR, or blank

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
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@OVOL	LB, SB[1]	Old volume	6 characters
EDG@OWN	GV	Owner to whom volume has been assigned	Character 8
	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	LD LV	Original expiration date	Date format
EDG@PACS	LC OPT	PREACS option	YES or NO
EDG@PDA	LC	PDA state	4 characters: ON, OFF, or NONE
EDG@PDAC	LC	PDA block count	3 characters: number 1 - 255
EDG@PDAL	LC	PDA log state	3 characters: ON or OFF
EDG@PDAS	LC	PDA block size	2 characters: numeric 1 - 31
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 S=SCRATCH R=REPLACE I=INIT E=ERASE N=NOTIFY O=OWNER	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, I
EDG@PID	LC, LR, SR[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

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
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	6 characters
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@RBYS	LV	Retained by set	3 characters: YES or NO
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
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@RTBY	LC	Retain by value	6 characters: VOLUME or SET
EDG@RTDJ	SV[1]	Retention date	Julian date format
EDG@RTDT	LD LV SV[1]	Retention date	Calendar date, CYCL/ccccc, PERMANENT, or WHILECATLG
EDG@RTM	LC	Time of last control data set report extract	8 characters (hh:mm:ss)
EDG@RUB	LC OPT	Reuse bin at	11 characters: one of CONFIRMMOVE, STARTMOVE
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@SCRM	LC VLPOOL [1]	Scratch mode	6 characters: AUTO or MANUAL

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@SDT	LV	Movement tracking date	Date format
EDG@SEC	LC[2]	Security classification number	Numeric: 0 - 255
EDG@SEQ	LV	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@SMP	LC	System-managed tape purge	One of YES, NO, ASIS
EDG@SMU	LC	System-managed tape update	One of 80 Exits, 40 Scratch, 20 Command, or N/A
EDG@SMUC	LC	System-managed tape command update	One of YES, NO
EDG@SMUE	LC	System-managed tape exit update	One of YES, NO
EDG@SMUS	LC	System-managed tape scratch update	One of YES, NO
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	8 characters (hh:mm:ss)
EDG@SRHN	LC OPTION	Server host name/IP address for a client	63 characters: 1 to 63 alphanumeric characters including hyphen, period, and blank
EDG@SRIP	LC OPTION	Server IP address	15 characters: 1 to 15 numeric characters including period and blank
EDG@SRPN	LC OPTION	Server port number	5 characters: 1 to 5 numeric characters. A number from 1 to 65535. 0 indicates there is no port number available.
EDG@SRTK	LC OPTION	Server tasks	3 characters: 1 to 3 numeric characters
EDG@SSTY	LC OPTION	Subsystem type	8 characters: CLIENT, SERVER, or STANDARD
EDG@SSM	LC	SMF security record number	Numeric: 0, 128 - 255; 0 is no security records written; 128 - 255 are the security record numbers

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@STEP	LD	Step name in job that created the data set	8 characters
EDG@STVC	LV, SV	Count of stacked volumes	Numeric: 10 digits maximum
EDG@SUR	LO	Owner's surname	20 characters
EDG@SYS	LV, SV[1]	Creation system ID	1-to-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, DSNAME, 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@VCAP	LV	Volume capacity	10 characters
EDG@VCHG	LC	VRSCHANGE 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	vvrrmm
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

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@VMIN	LC	VRSMIN count value	A number
EDG@VMV	LD	Vital record specification management value	8 characters, defined by your installation
EDG@VNDR	LV VOL, SV(e)	Vendor information	8 alphanumeric or special characters
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] SV[1]	Volume serial number	6 characters
EDG@VOLT	LV	Volume type	One of LOGICAL, PHYSICAL, or STACKED
EDG@VOL1	LV	VOL1 label volser	6 characters
EDG@VPCT	LV	Volume percentage full	0 - 100
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	8 characters (hh:mm:ss)
EDG@VTYP	LD	Primary vital record specification type	DATASET, SMSMC, VRSMV, DSN/MV, or DSN/MC
EDG@VWMC	LV STAT, SV(e)	Volume write mount count	Numeric: 0 - 99999
EDG@WWID	LV, SV(e)	World wide identifier	24 hexadecimal characters
EDG@XDC	LC[1]	Expiration date check	One of Y, N, or O

Table 43. TSO Subcommand Variables by Name (continued)

Variable Name	Subcommands	Contents	Format
EDG@XDT	LC	Date of last inventory management expiration processing	Date format
	LD SD	Data set expiration date	Date format
	LV SV[1]	Volume expiration date	Date format
EDG@XTM	LC	Time of last inventory management expiration processing	8 characters (hh:mm:ss)
EDG@X100	LC	EDGUX100 installation exit status	8 characters: ENABLED, DISABLED, or NONE
EDG@X200	LC	EDGUX200 installation exit status	8 characters: ENABLED, DISABLED, or NONE
EDG@2JBN	LD	Secondary vital record specification jobname mask	8 characters
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

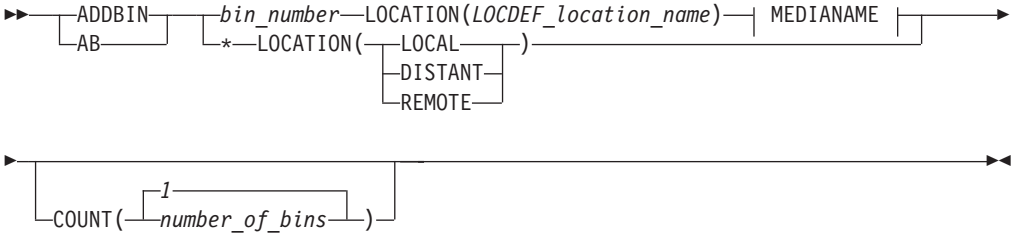
Notes:

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 363.
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.

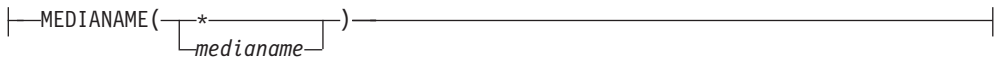
Appendix A. DFSMSrmm TSO Subcommands

The IBM tape management product DFSMSrmm provides the RMM TSO subcommand and a set of subcommands to request DFSMSrmm functions.

ADDBIN: Adding a Bin Number in a Storage Location

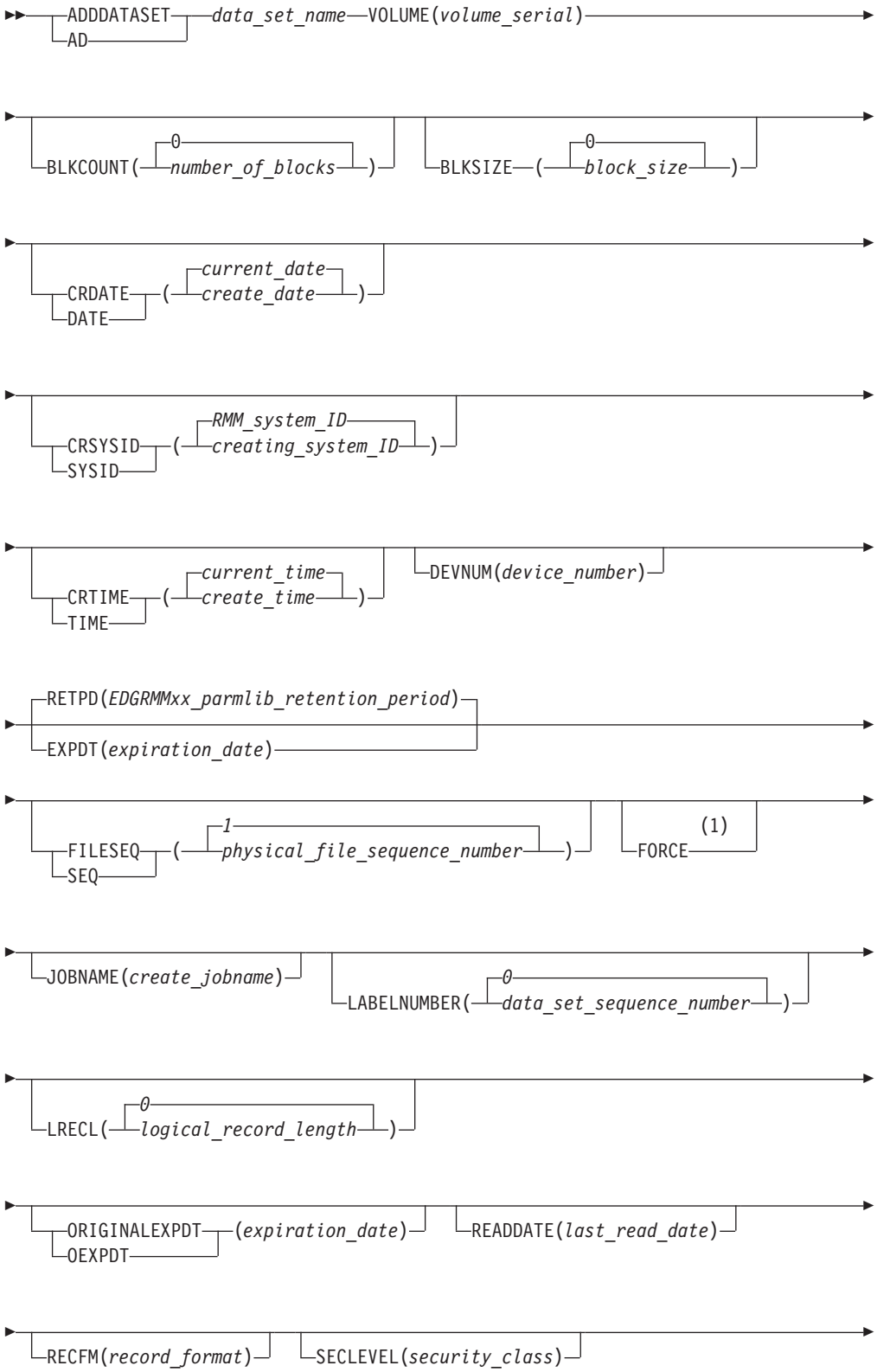


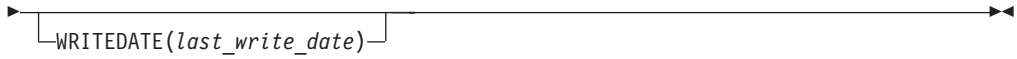
MEDIANAME:



© Copyright IBM Corp. 1992, 2004

ADDDATASET: Adding Data Set Information

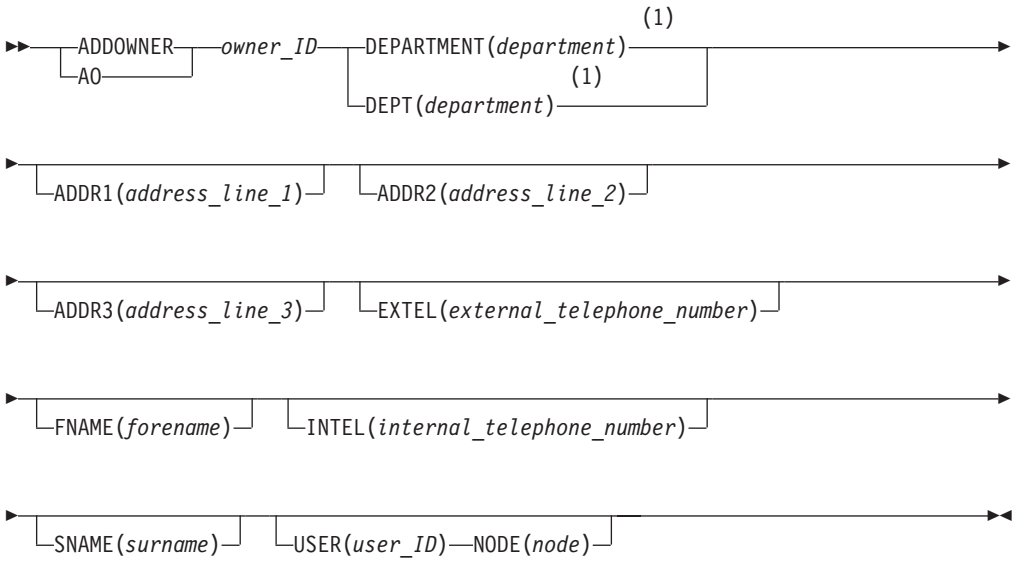




Notes:

1 Use the FORCE operand, which requires CONTROL access to the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource, to add a data set to a volume that was recorded by DFSMSrmm during O/C/EOV processing,.

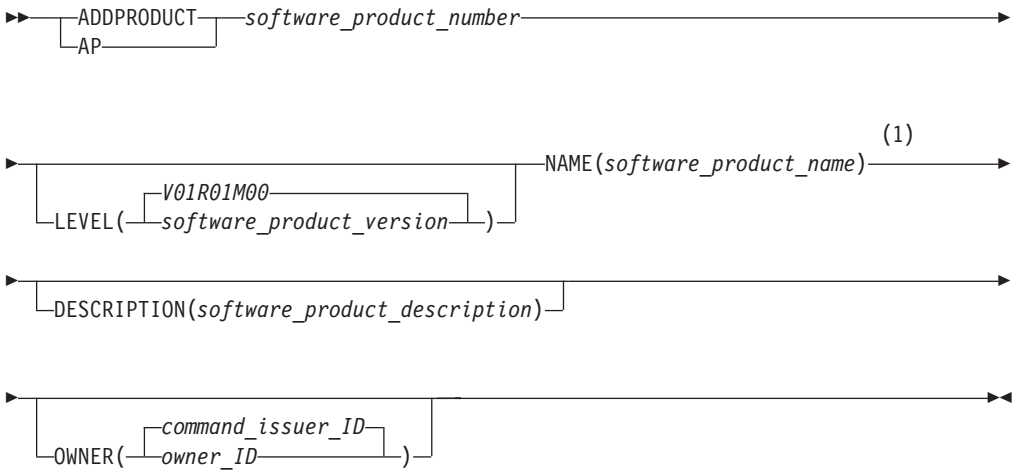
ADDOWNER: Adding Owner Information



Notes:

1 The DEPARTMENT operand must contain at least one non-blank character.

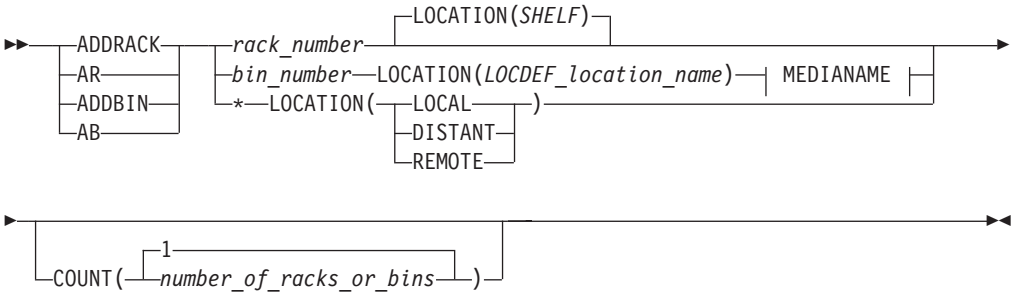
ADDPRODUCT: Adding Software Product Information



Notes:

1 The NAME operand must contain at least one non-blank character.

ADDRACK: Adding a Shelf Location



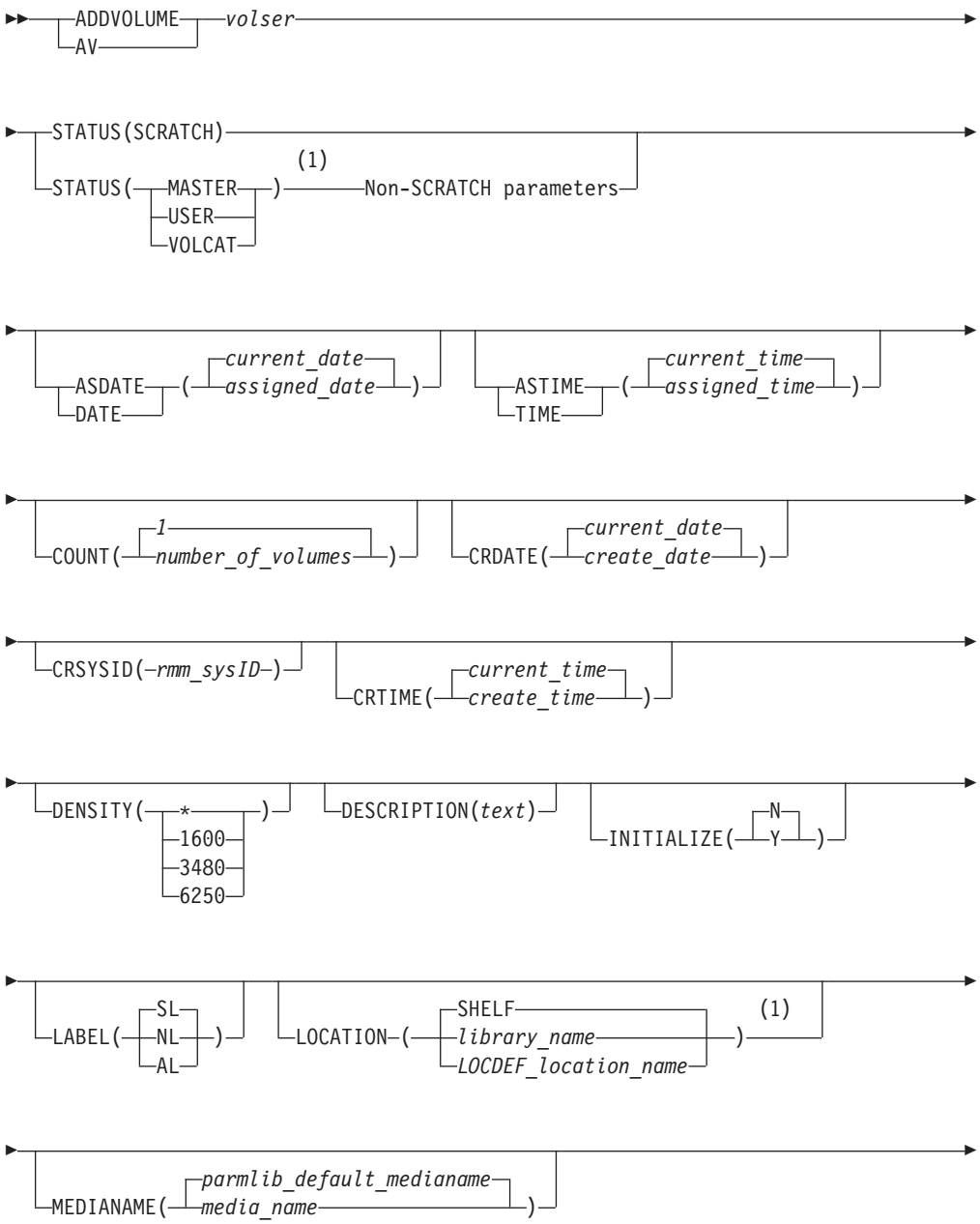
MEDIANAME:

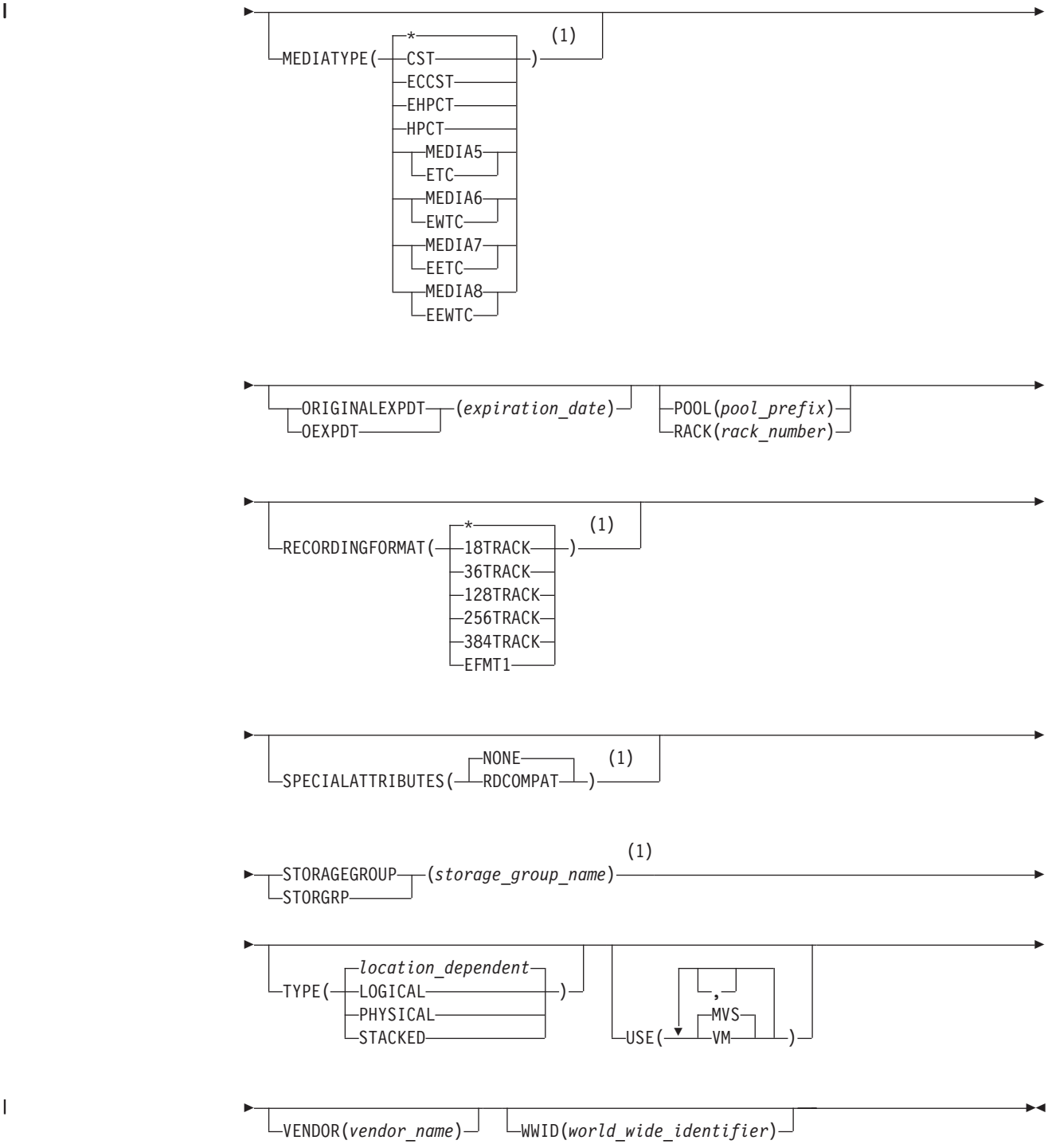
ADDVOLUME: Adding Volume Information

The ADDVOLUME operands are broken down into multiple syntax diagrams:

- ADDVOLUME: Volume Operands
- ADDVOLUME: Non-SCRATCH Volume Optional Operands

ADDVOLUME: Volume Operands

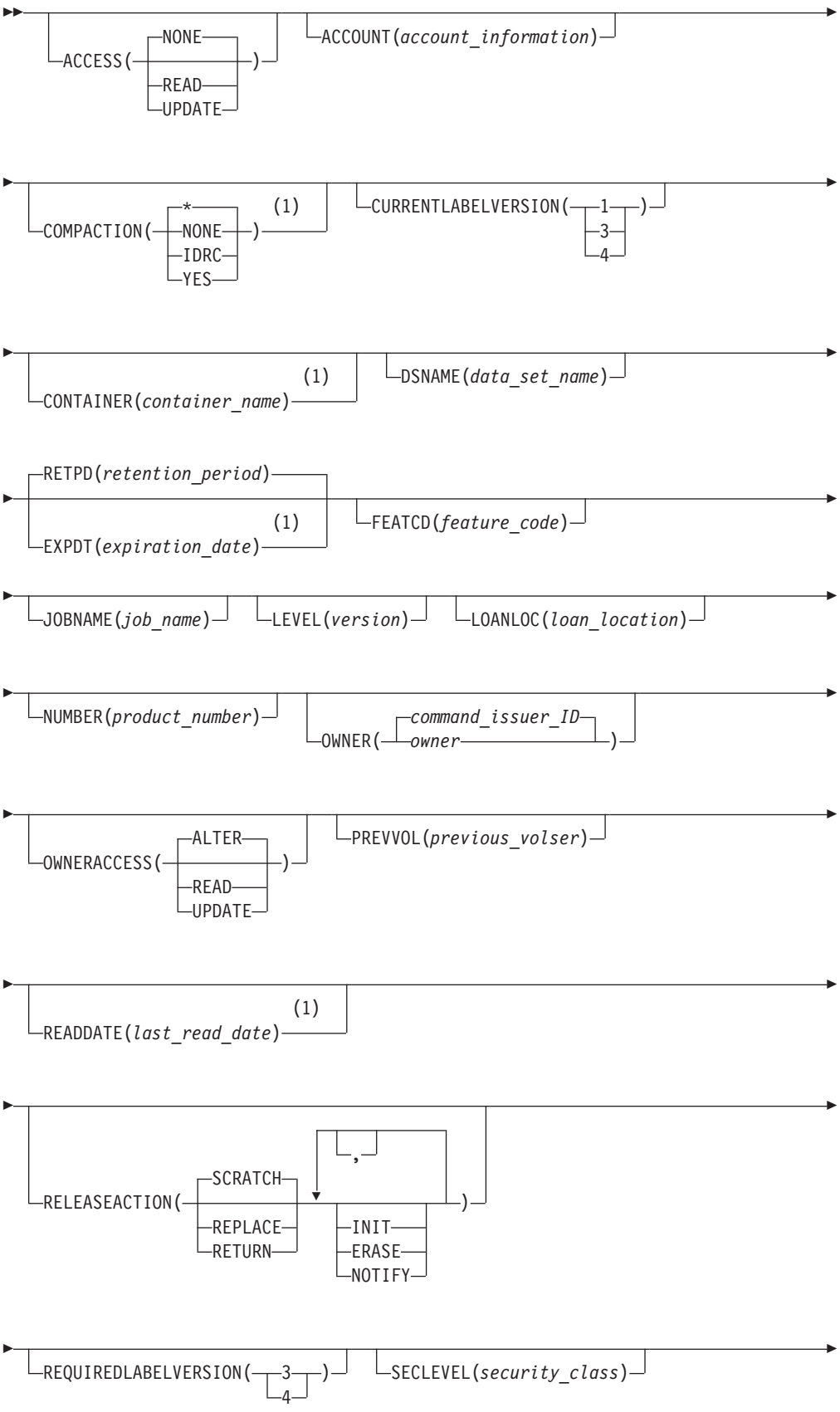


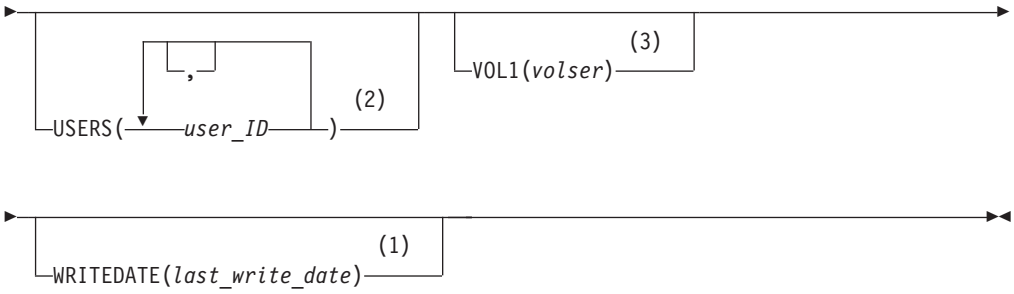


Notes:

1 This operand uses the tape configuration database information when STATUS(VOLCAT) is specified.

ADDVOLUME: Non-SCRATCH Volume Optional Operands


ACCESS(NONE READ UPDATE)
ACCOUNT(account_information)
COMPACTION(* NONE IDRC YES) (1)
CURRENTLABELVERSION(1 3 4)
CONTAINER(container_name) (1)
DSNAME(data_set_name)
RETPD(retention_period)
EXPDT(expiration_date) (1)
FEATCD(feature_code)
JOBNAME(job_name)
LEVEL(version)
LOANLOC(loan_location)
NUMBER(product_number)
OWNER(command_issuer_ID owner)
OWNERACCESS(ALTER READ UPDATE)
PREVVOL(previous_volser)
READDATE(last_read_date) (1)
RELEASEACTION(SCRATCH REPLACE RETURN , INIT ERASE NOTIFY)
REQUIREDLABELVERSION(3 4)
SECLEVEL(security_class)



Notes:

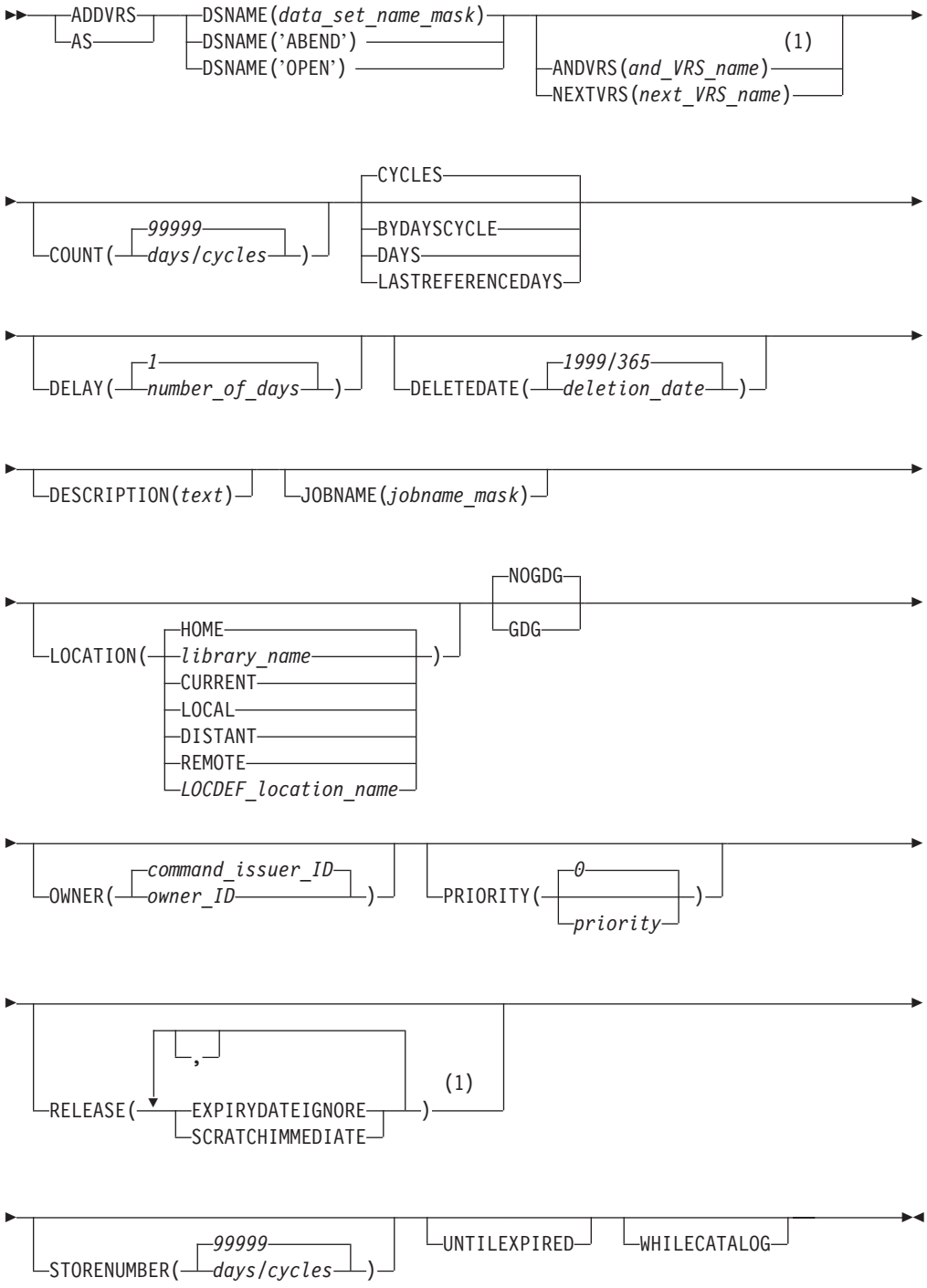
1 This operand uses the tape configuration database information when STATUS(VOLCAT) is specified.

2 You can specify a maximum of 12 user IDs.

3 This operand cannot be specified for SCRATCH volumes, LOGICAL volumes, STACKED volumes, or NOLABEL volumes.

ADDVRS: Adding a Vital Record Specification

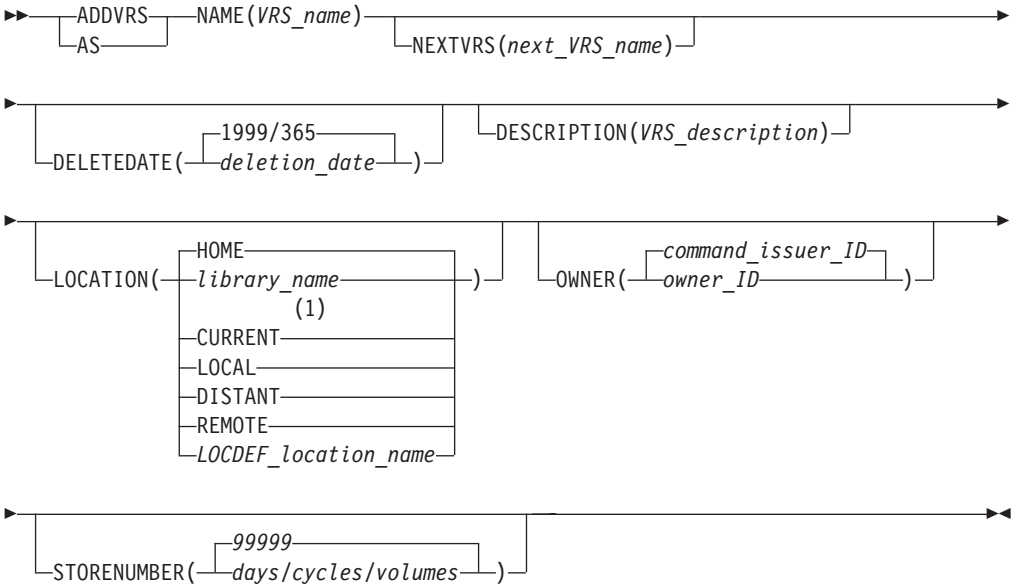
ADDVRS: Adding a Data Set Vital Record Specification



Notes:

1 This operand is honored when you specify the DFSMSrmm parmlib OPTION VRSEL(NEW) operand.

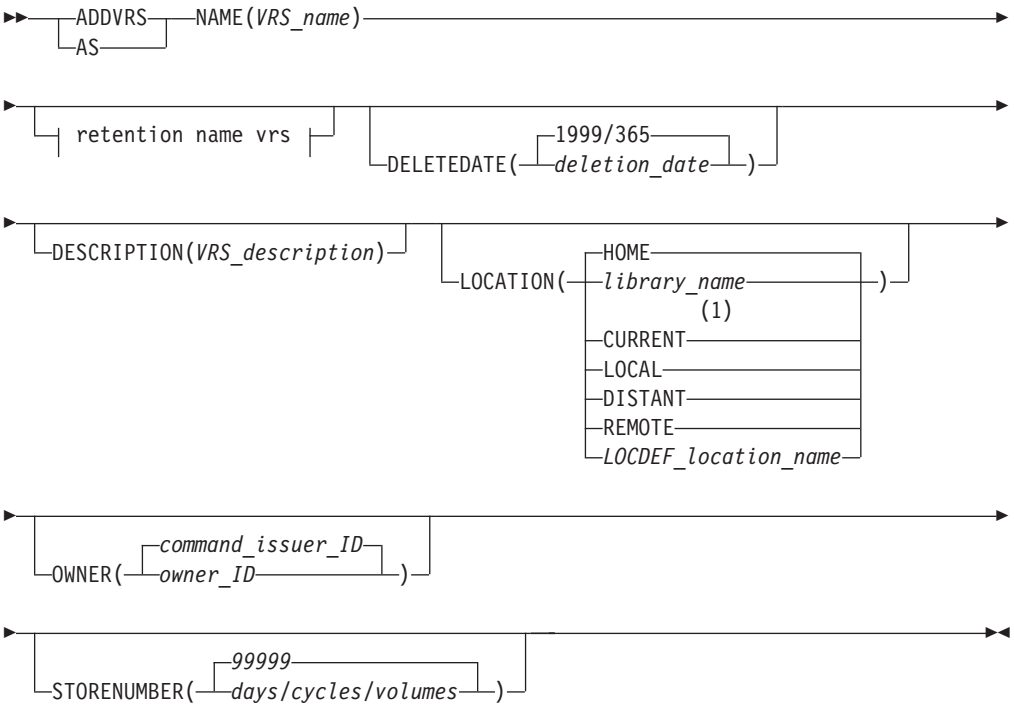
ADDVRS: Adding a Location Name Vital Record Specification



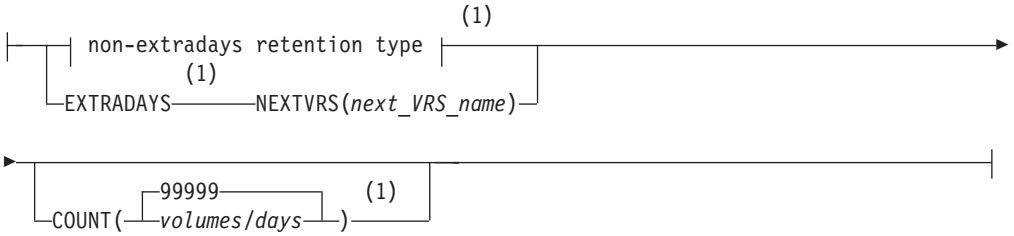
Notes:

1 This operand is honored when you specify the DFSMSrmm parmlib OPTION VRSEL(NEW) operand.

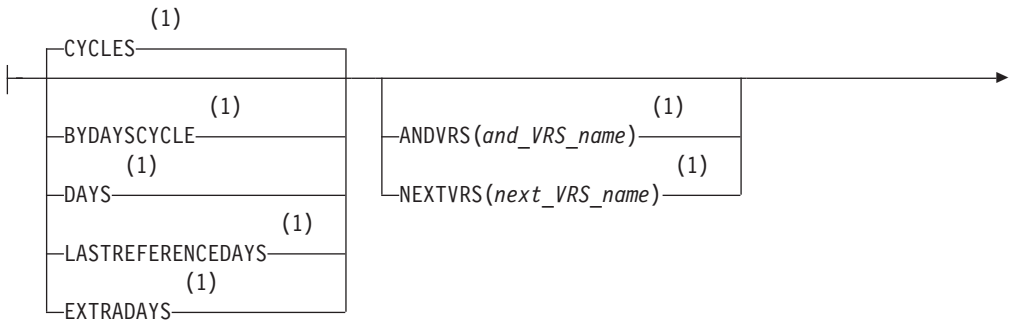
ADDVRS: Adding a Retention Name Vital Record Specification

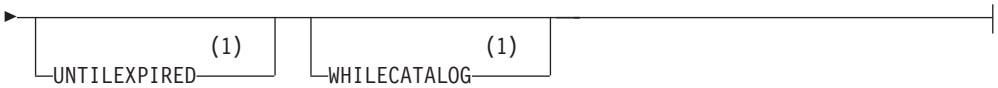


retention name vrs:



non-extradays retention type:

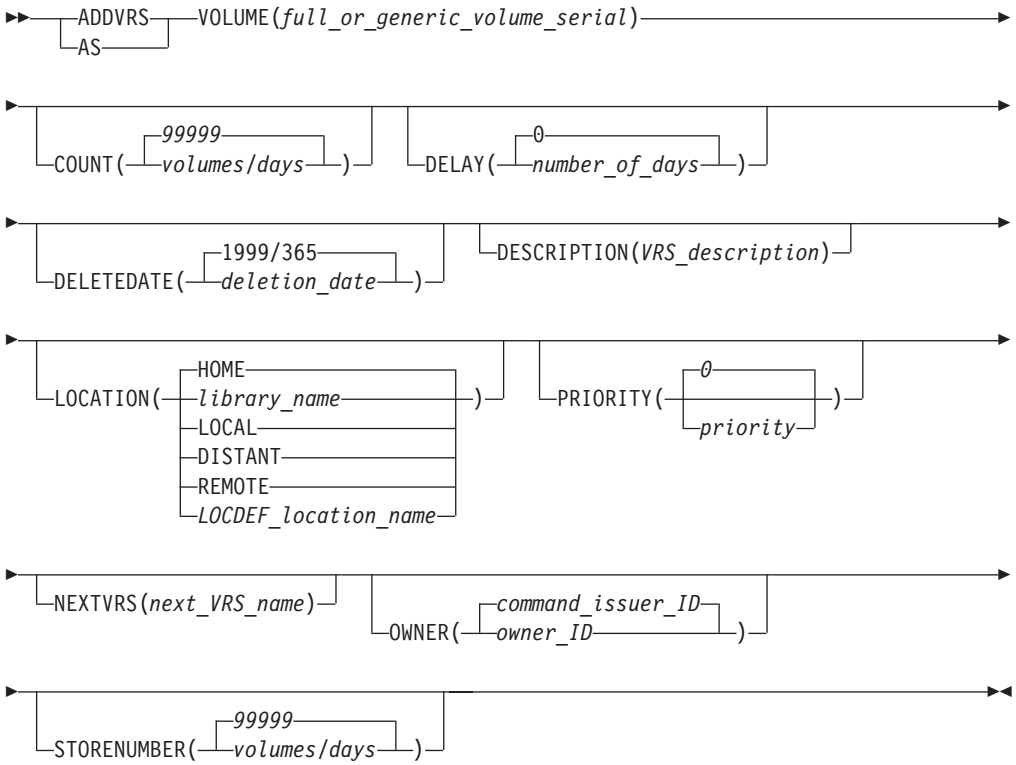




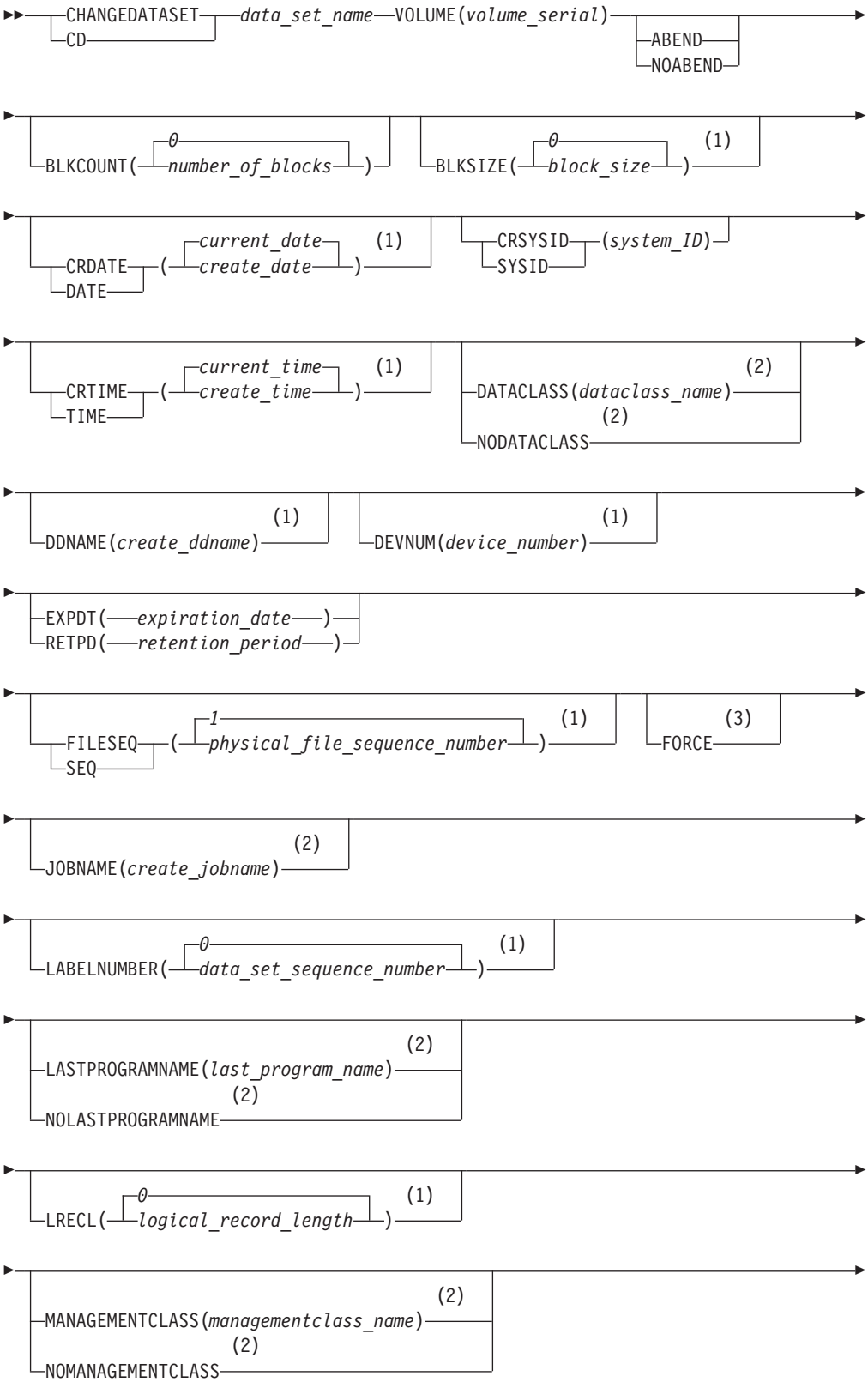
Notes:

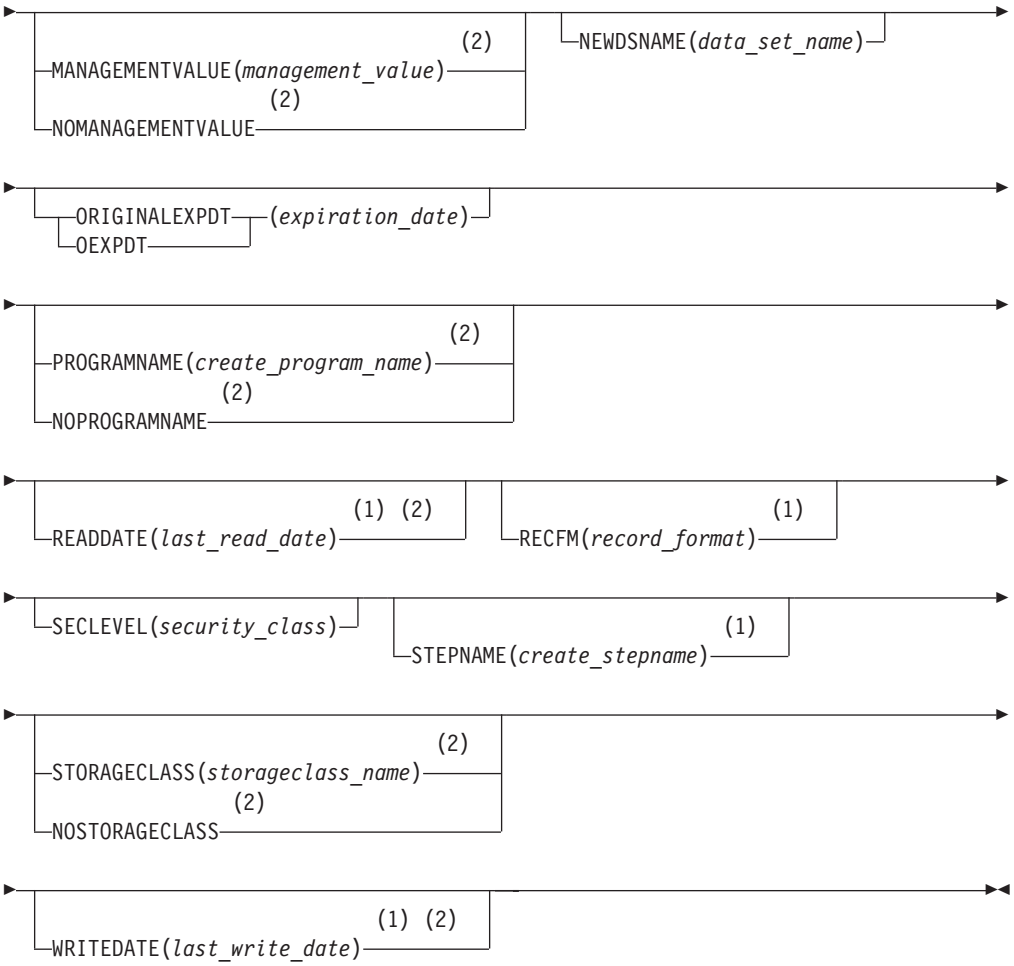
1 This operand is honored when you specify the DFSMSrmm parmlib OPTION VRSEL(NEW) operand. When you run inventory management VRSEL(OLD), DFSMSrmm issues return code 4.

ADDVRS: Adding a Volume Vital Record Specification



CHANGEDATASET: Changing Data Set Information





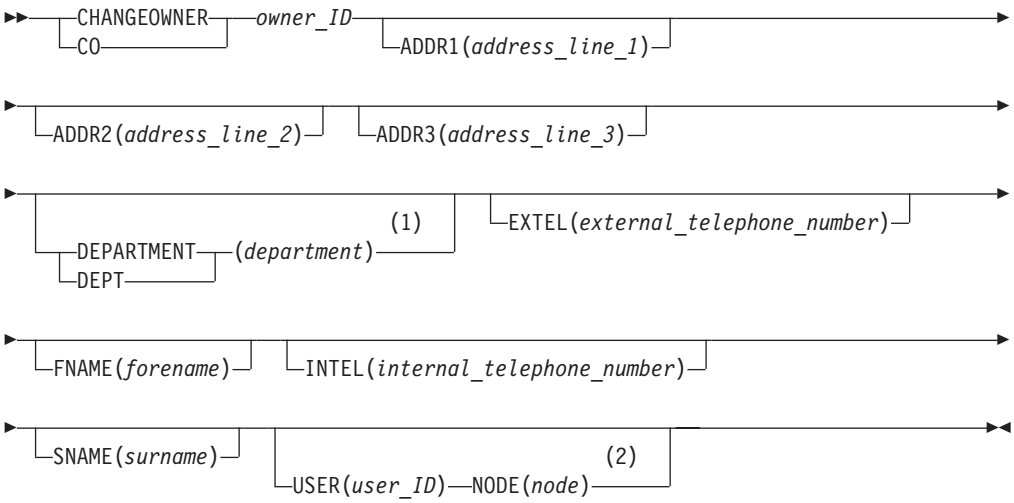
Notes:

1 This operand cannot be specified if DFSMSrmm has automatically recorded information about the volume when a data set on the volume was first opened.

2 This operand can only be specified by a user with CONTROL access to the STGADMIN.EDG.MASTER security resource. Owner authorization does not apply.

3 Specify this operand 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 the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource to use the FORCE operand.

CHANGEOWNER: Changing Owner Information

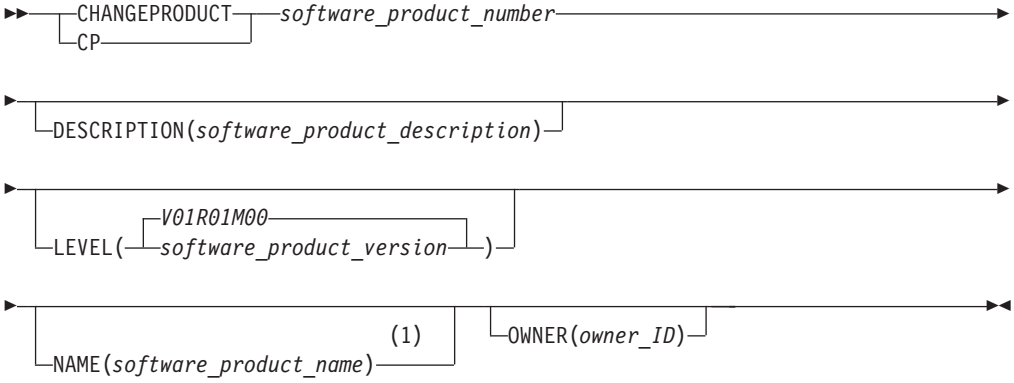


Notes:

1 The DEPARTMENT operand must contain at least one non-blank character.

2 The values you specify for USER(*user_ID*) and NODE(*node*) work together; if you delete one, you must also delete the other.

CHANGEPRODUCT: Changing Software Product Information



Notes:

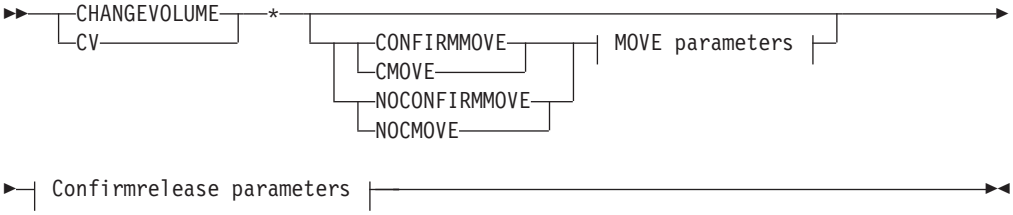
1 The NAME operand must contain at least one non-blank character.

CHANGEVOLUME: Changing Volume Information

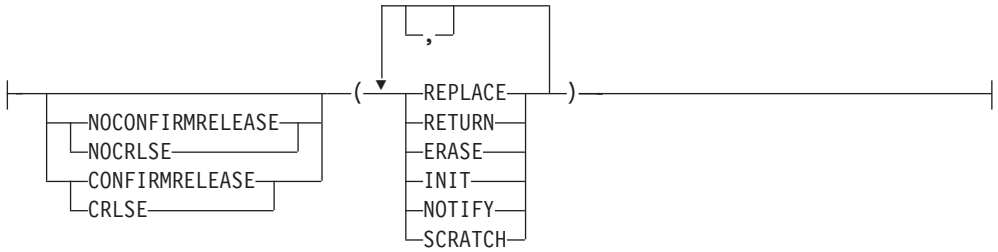
The CHANGEVOLUME operands are broken down into multiple syntax diagrams:

- CHANGEVOLUME: Confirmrelease and Global Movement Operands
- CHANGEVOLUME: Specific Volume Optional Operands which spans multiple pages.

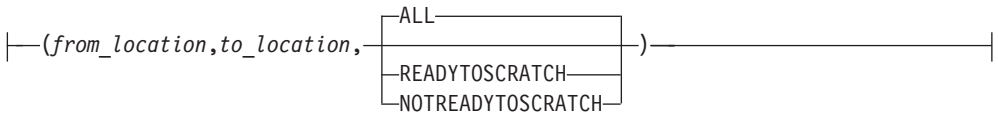
CHANGEVOLUME: Confirmrelease and Global Movement Operands



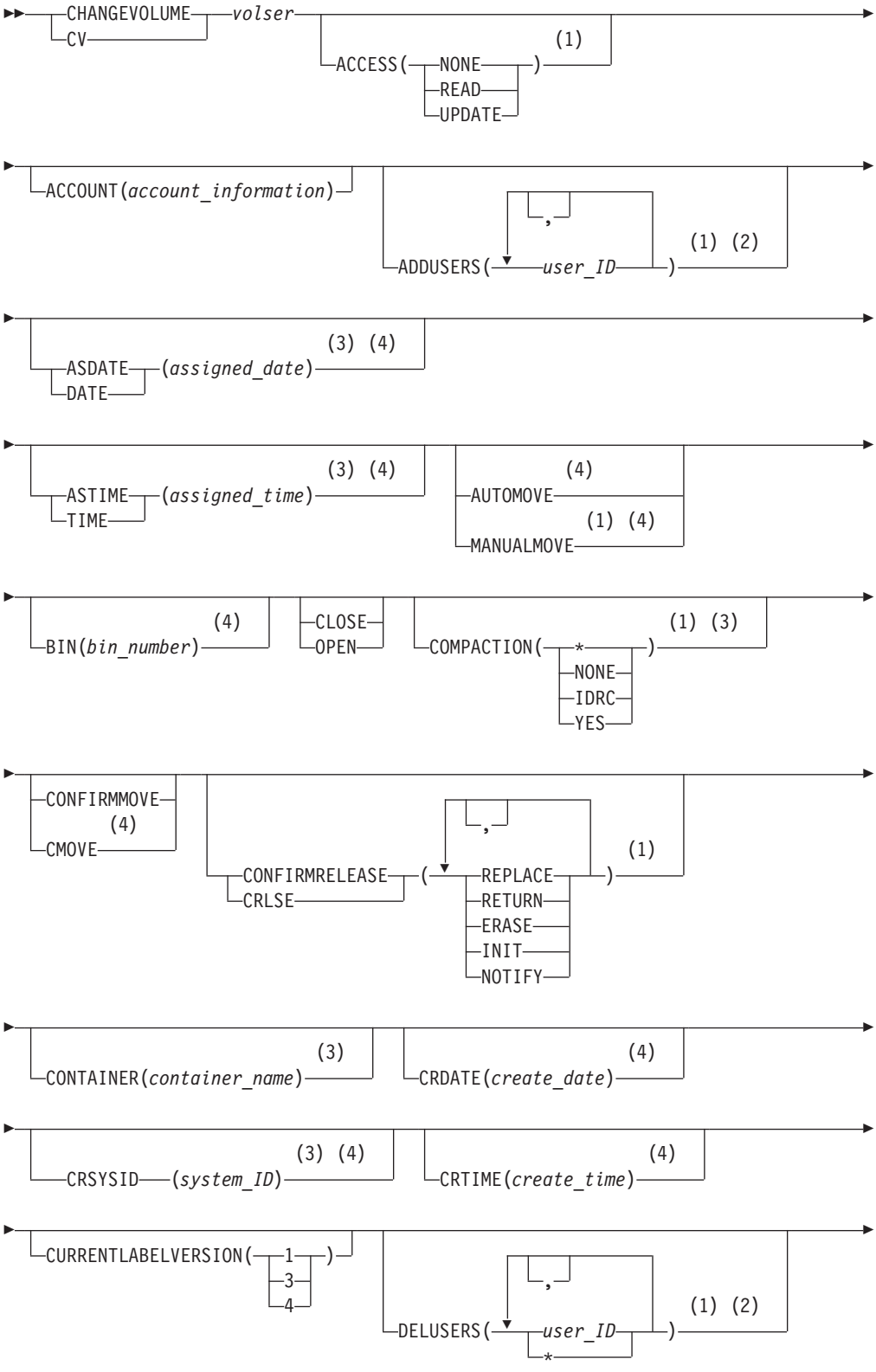
Confirmrelease parameters:

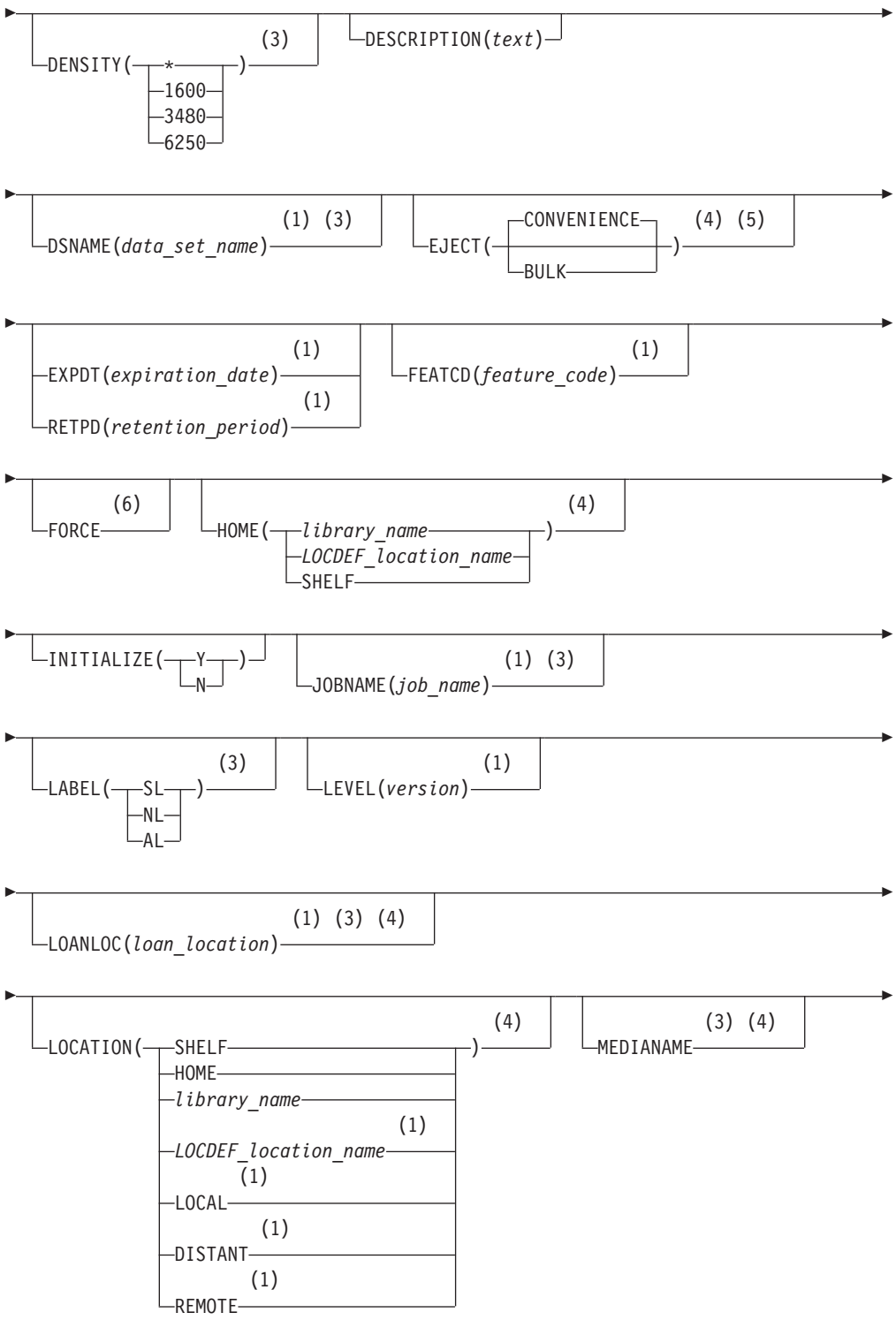


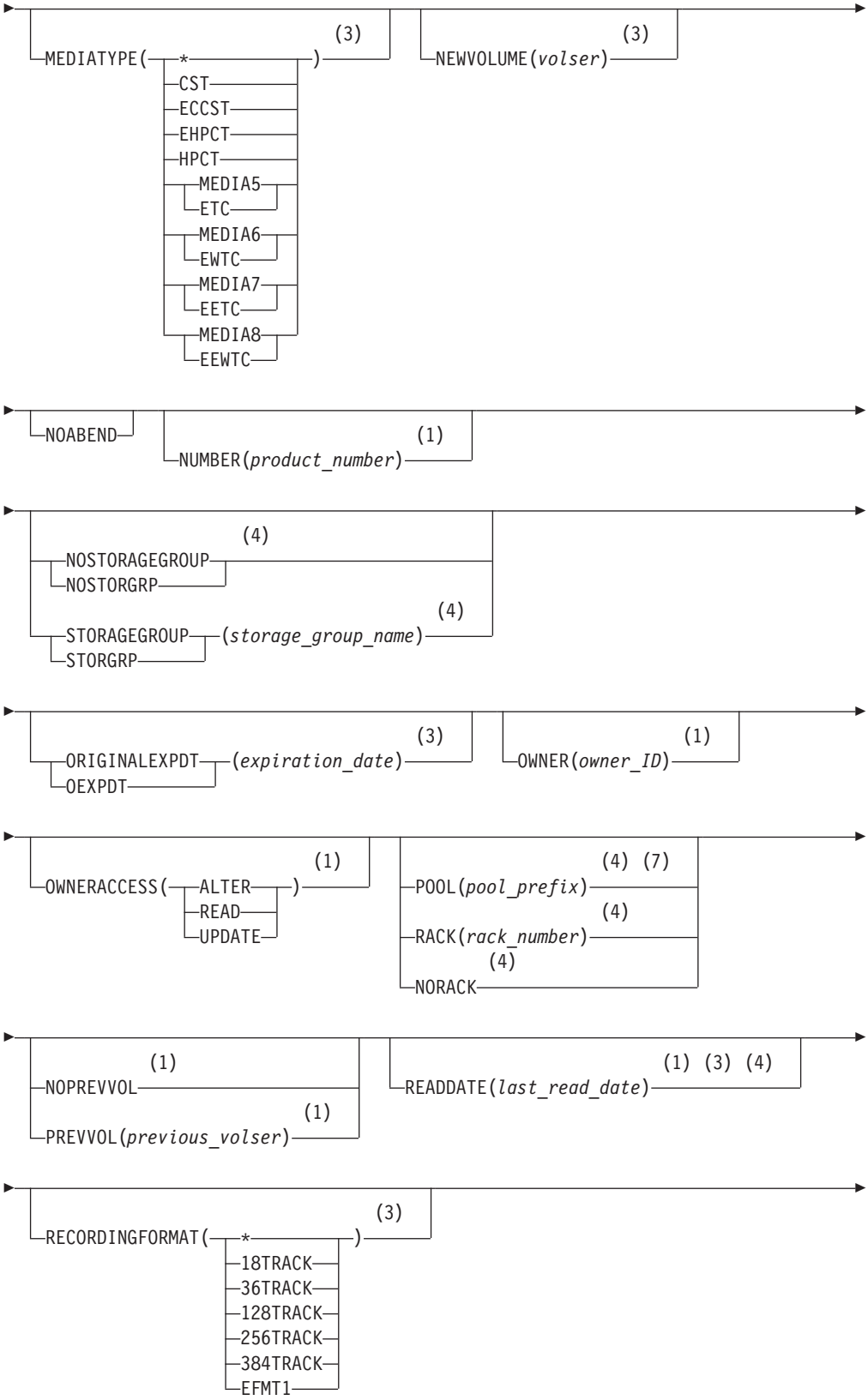
MOVE parameters:

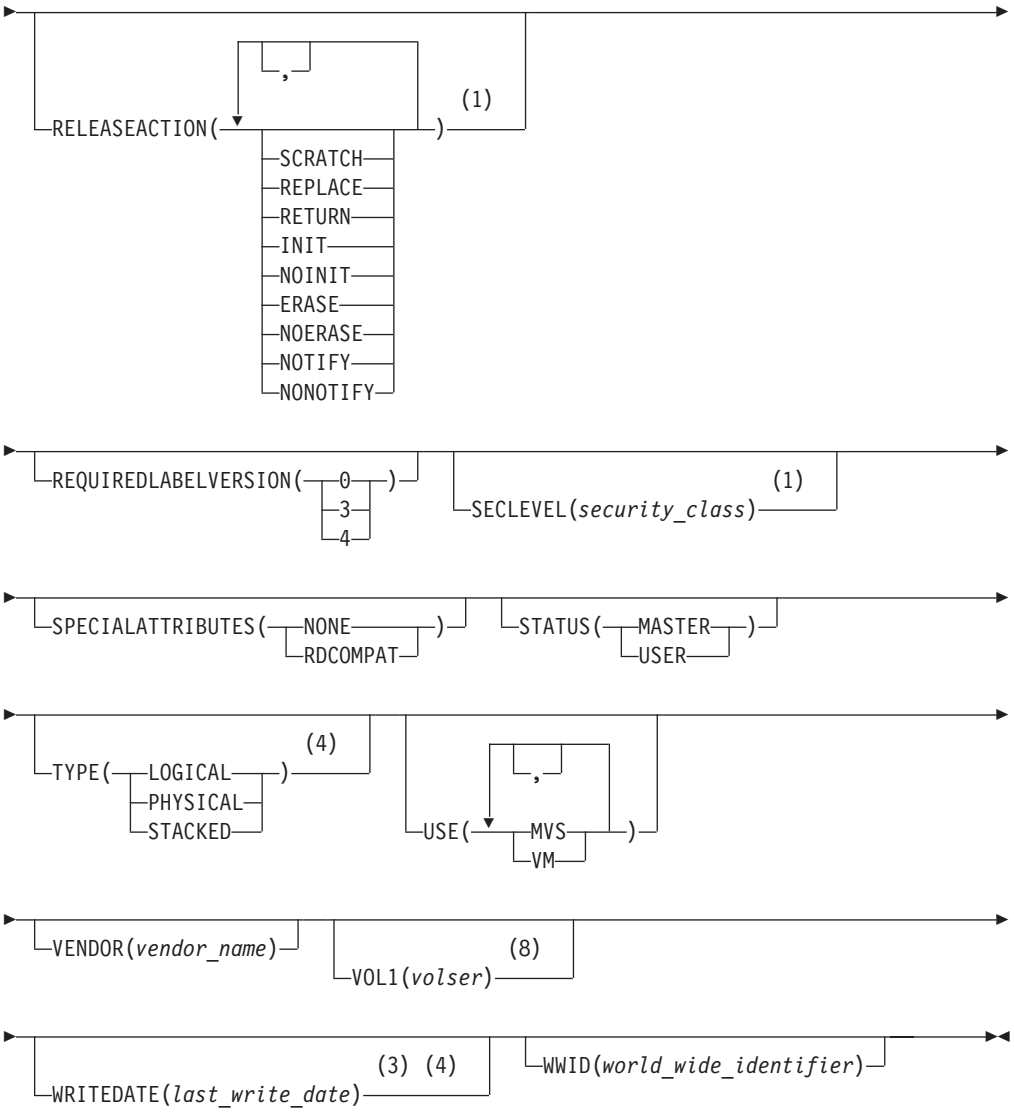


CHANGEVOLUME: Specific Volume Optional Operands




DENSITY(*) (3)
1600
3480
6250
DESCRIPTION(text)
DSNAME(data_set_name) (1) (3)
EJECT(CONVENIENCE BULK) (4) (5)
EXPDT(expiration_date) (1)
RETPD(retention_period) (1)
FEATCD(feature_code) (1)
FORCE (6)
HOME(library_name LOCDEF_location_name SHELF) (4)
INITIALIZE(Y N)
JOBNAME(job_name) (1) (3)
LABEL(SL NL AL) (3)
LEVEL(version) (1)
LOANLOC(loan_location) (1) (3) (4)
LOCATION(SHELF HOME library_name LOCDEF_location_name (1) LOCAL (1) DISTANT (1) REMOTE) (4)
MEDIANAME (3) (4)


MEDIATYPE(*) (3)
CST
ECCST
EHPCT
HPCT
MEDIA5
ETC
MEDIA6
EWTC
MEDIA7
EETC
MEDIA8
EEWTC
NEWVOLUME(volser) (3)
NOABEND
NUMBER(product_number) (1)
NOSTORAGEGROUP (4)
NOSTORGRP
STORAGEGROUP(storage_group_name) (4)
STORGRP
ORIGINALEXPDT(expiration_date) (3)
OEXPDT
OWNER(owner_ID) (1)
OWNERACCESS(ALTER) (1)
READ
UPDATE
POOL(pool_prefix) (4) (7)
RACK(rack_number) (4)
NORACK (4)
NOPREVVOL (1)
PREVVOL(previous_volser) (1)
READDATE(last_read_date) (1) (3) (4)
RECORDINGFORMAT(*) (3)
18TRACK
36TRACK
128TRACK
256TRACK
384TRACK
EFMT1



Notes:

1 This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

2 You can specify a maximum of 12 user IDs.

3 You can specify this operand only if you also specify the FORCE operand. To use the FORCE operand, you must have CONTROL access to the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource.

4 This operand can be specified if the user has CONTROL access to STGADMIN.EDG.MASTER resource.

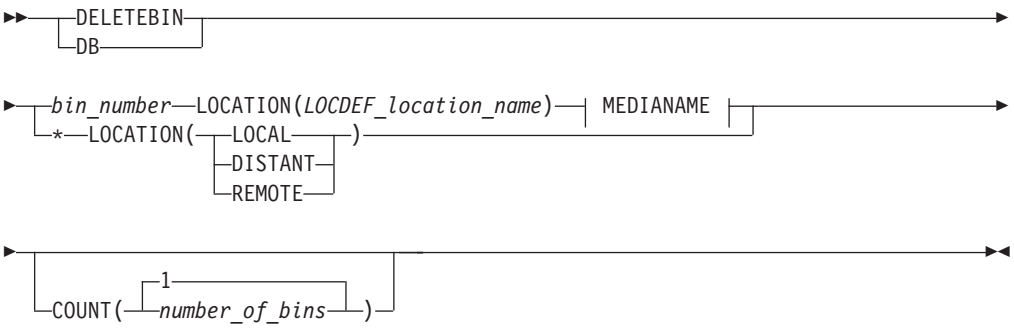
5 This operand can be specified only if the volume resides in a system-managed library.

6 This operand 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 the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource to use the FORCE operand.

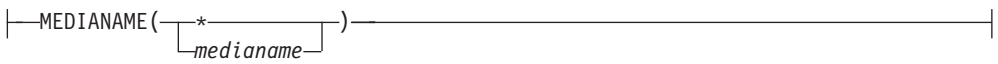
7 This operand cannot be specified if the volume resides in an IBM Tape Library Dataserver.

8 This operand cannot be specified for SCRATCH volumes, LOGICAL volumes, STACKED volumes, or NOLABEL volumes.

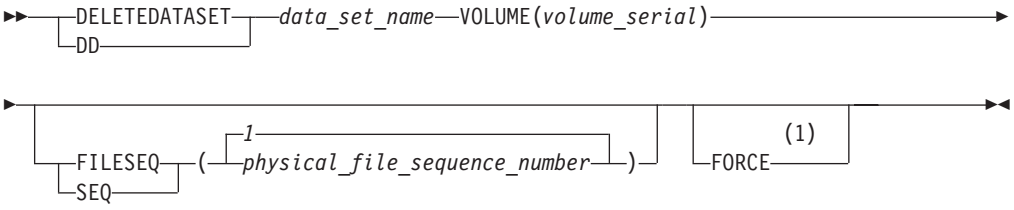
DELETEBIN: Deleting Bin Number Information



MEDIANAME:



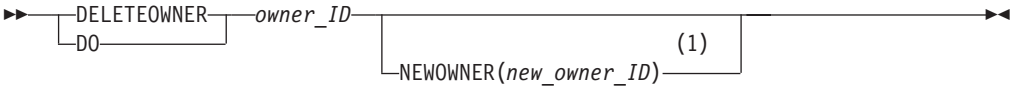
DELETEDATASET: Deleting Data Set Information



Notes:

1 Specify the FORCE operand to delete a data set for a volume where information was recorded by DFSMSrmm during O/C/EOV processing. You must have CONTROL access to the STGADMIN.EDG.MASTER security resource and UPDATE access to the STGADMIN.EDG.FORCE security resource to use the FORCE operand.

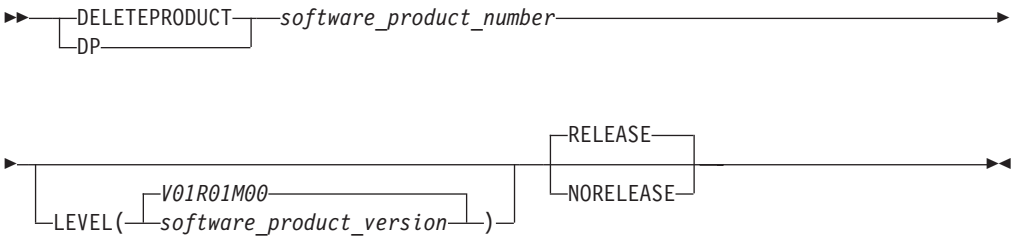
DELETEOWNER: Deleting Owner Information



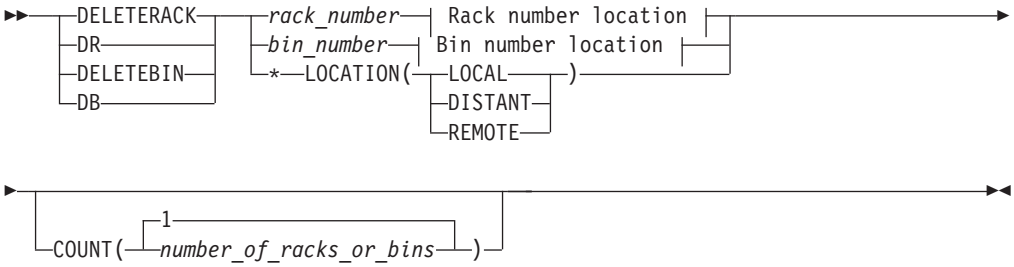
Notes:

1 The NEWOWNER operand must be specified to reassign volumes if the owner you are deleting owns one or more volumes.

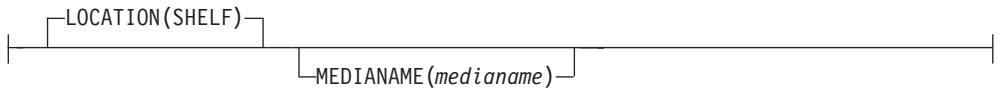
DELETEPRODUCT: Deleting Software Product Information



DELETERACK: Deleting Shelf Location Information

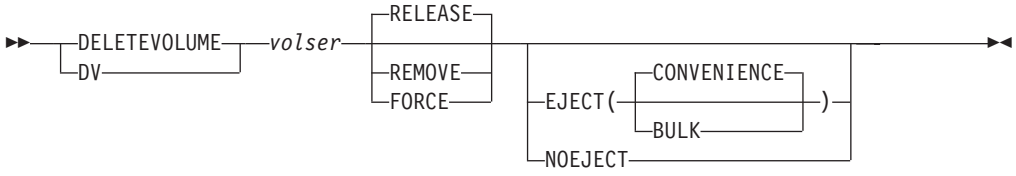


Rack number location:

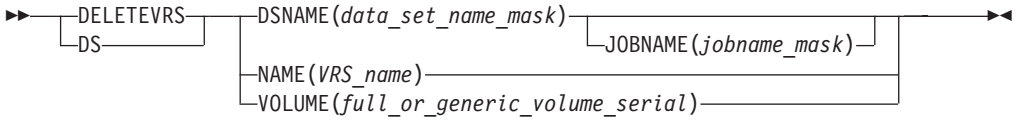


Bin number location:

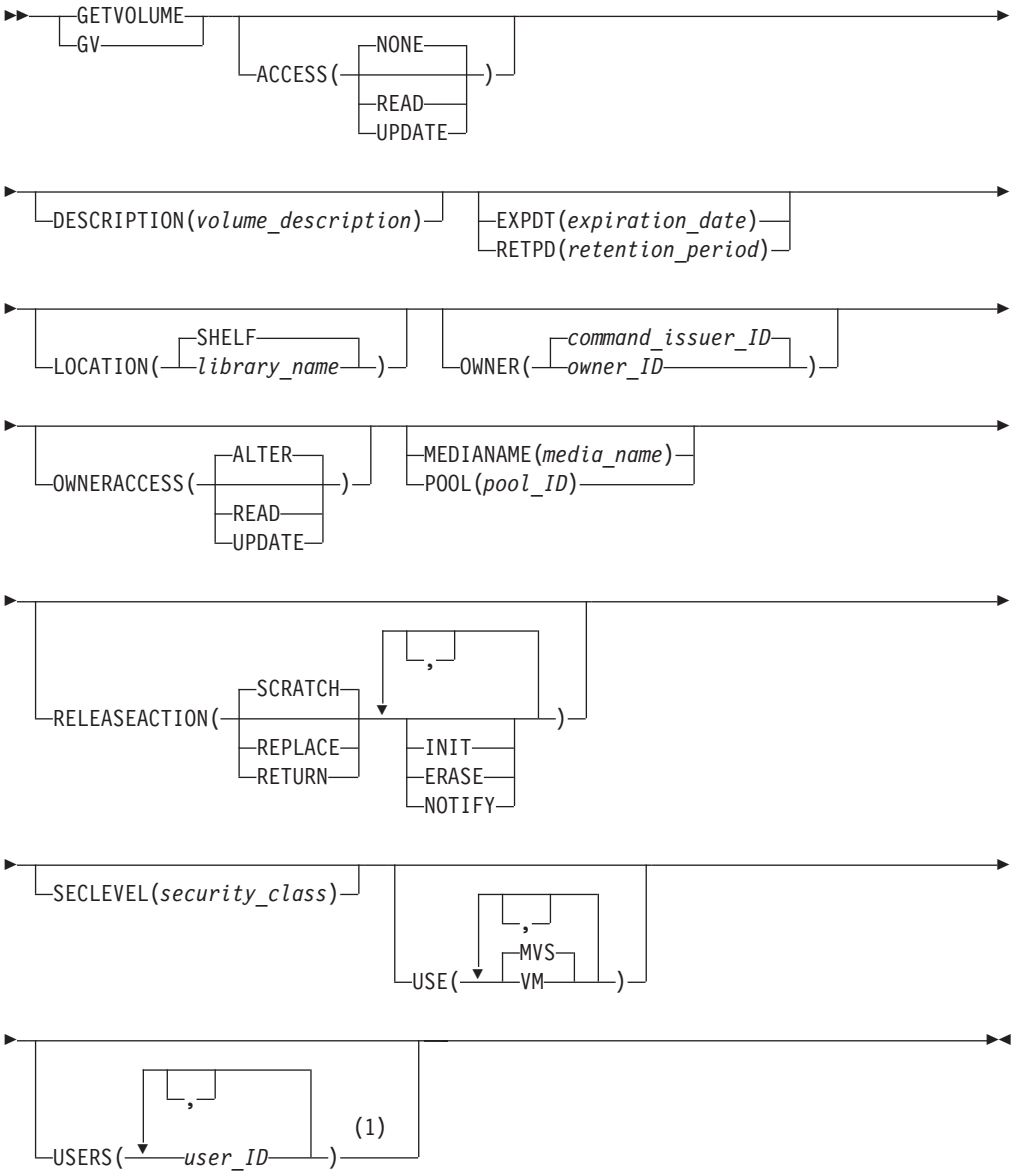
DELETEVOLUME: Deleting Volume Information



DELETEVRS: Deleting Vital Record Specification Information



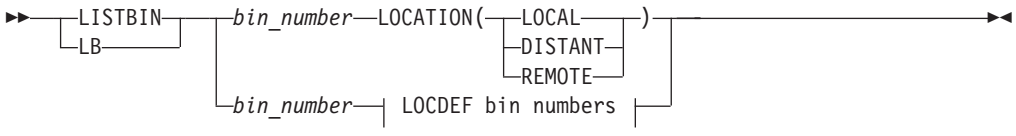
GETVOLUME: Requesting and Assigning Scratch Volumes



Notes:

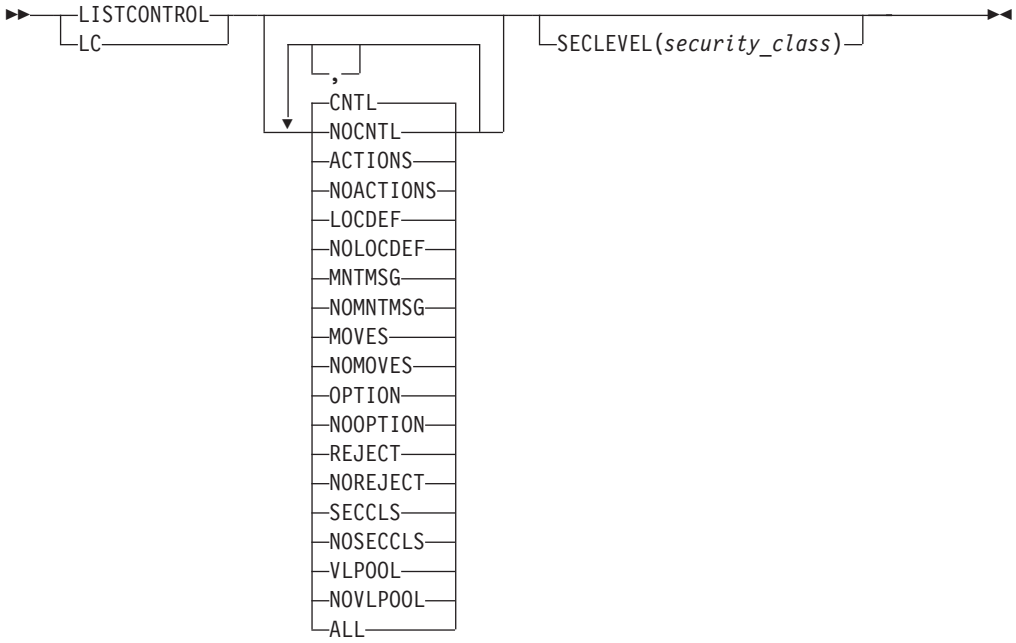
1 You can specify a maximum of 12 user IDs.

LISTBIN: Displaying Information about a Shelf Location

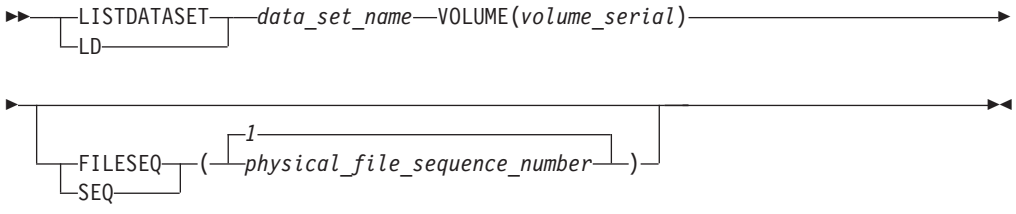


LOCDEF bin numbers:

LISTCONTROL: Displaying Parmlib Options and Control Information

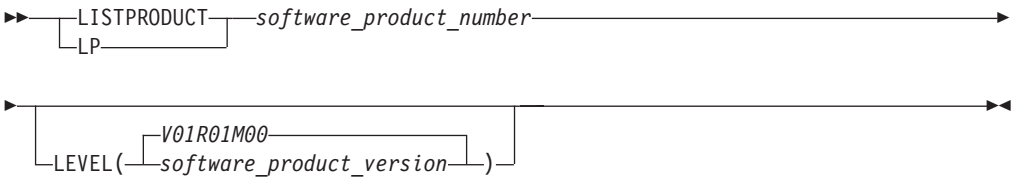


LISTDATASET: Displaying Information about a Data Set

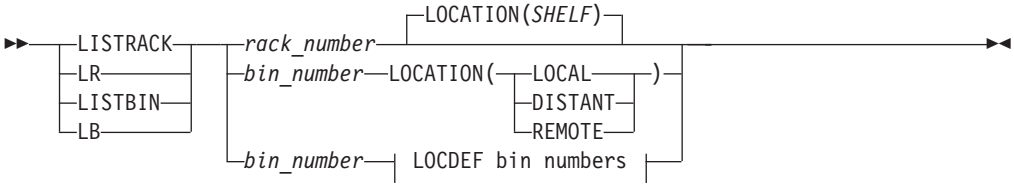


LISTOWNER: Displaying Information about an Owner

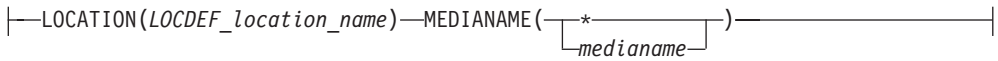
LISTPRODUCT: Displaying Information about a Software Product



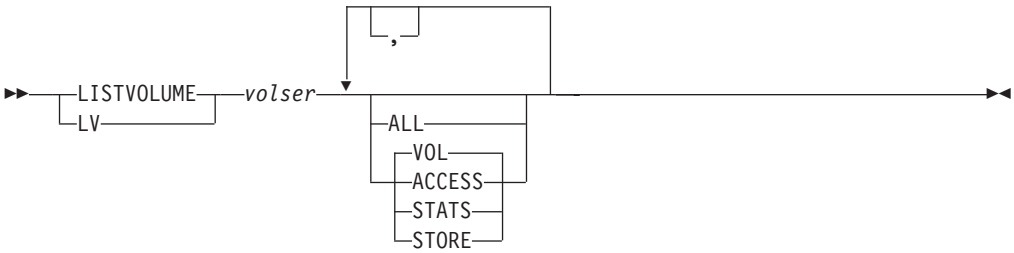
LISTRACK: Displaying Information about a Shelf Location



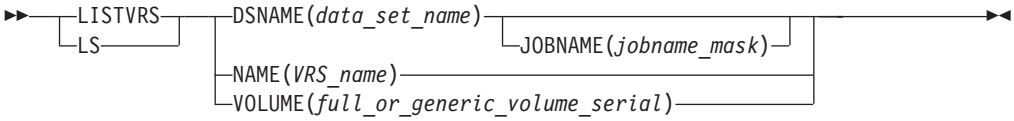
LOCDEF bin numbers:



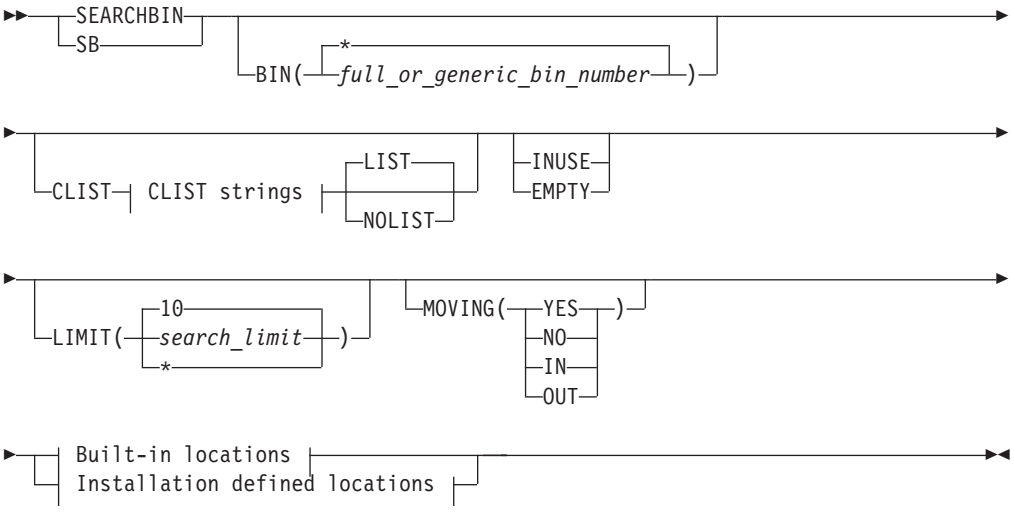
LISTVOLUME: Displaying Information about a Volume



LISTVRS: Displaying Information about a Vital Record Specification



SEARCHBIN: Creating a List of Bin Numbers

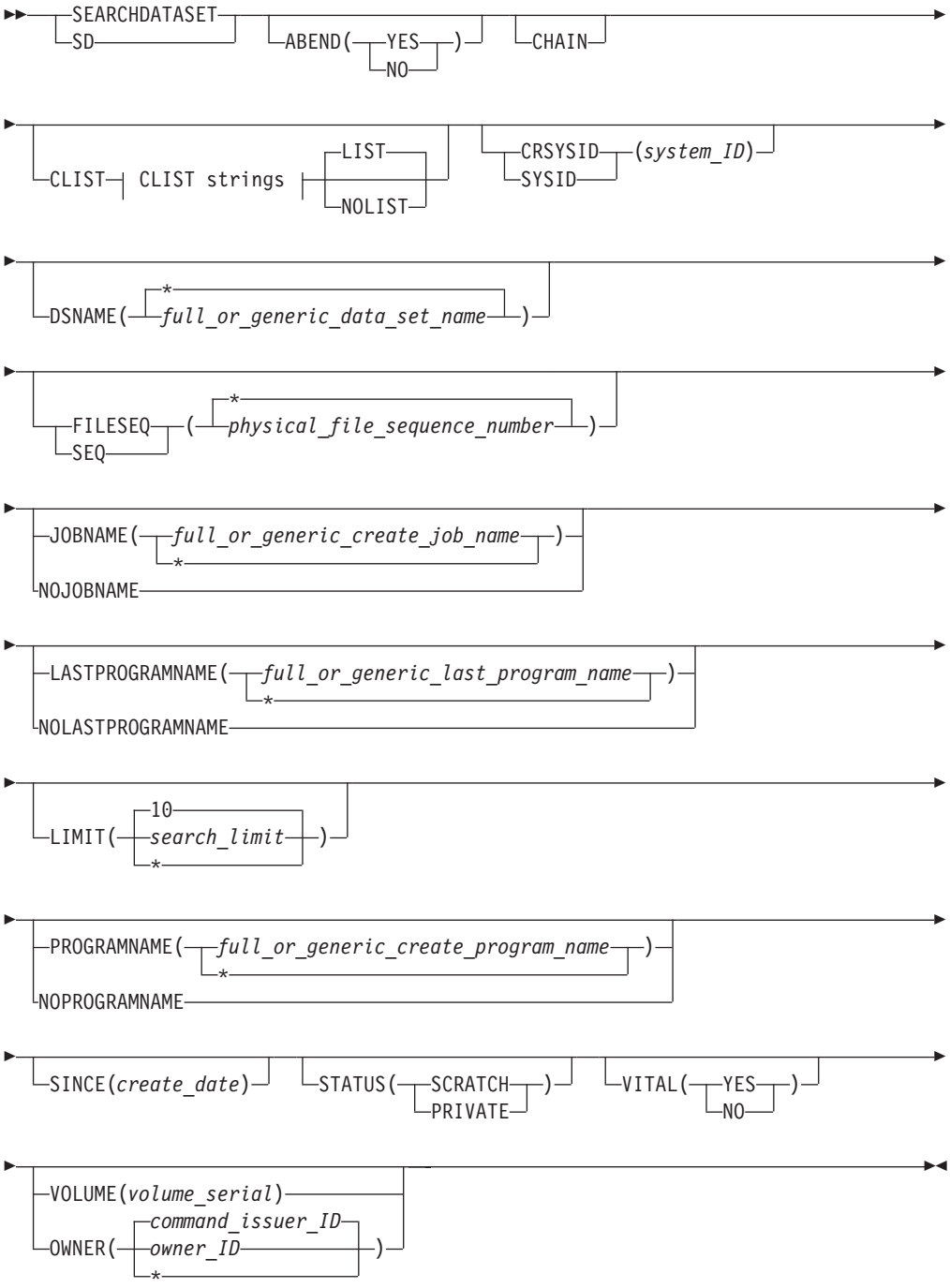


CLIST strings:

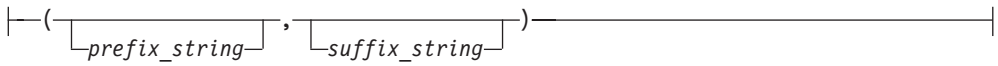
Built-in locations:

Installation defined locations:

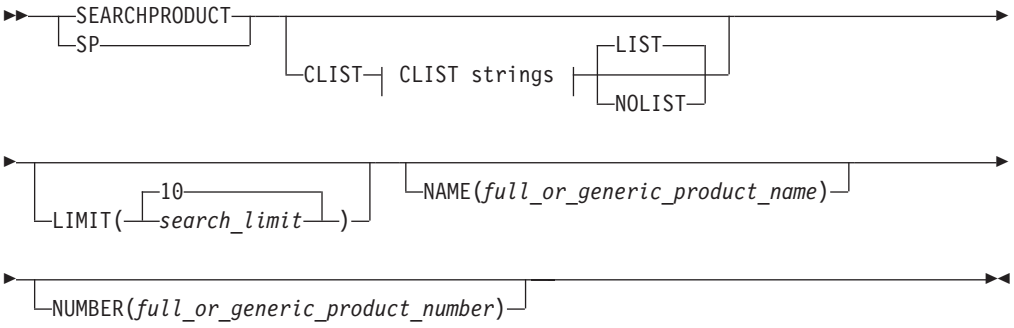
SEARCHDATASET: Creating a List of Data Sets



CLIST strings:

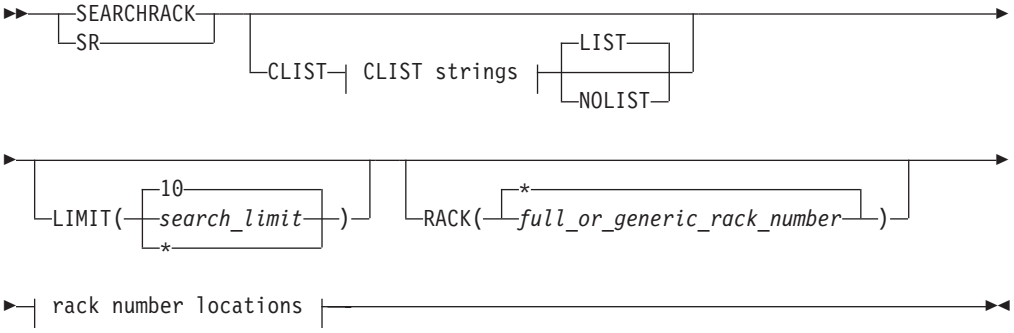


SEARCHPRODUCT: Creating a List of Software Products



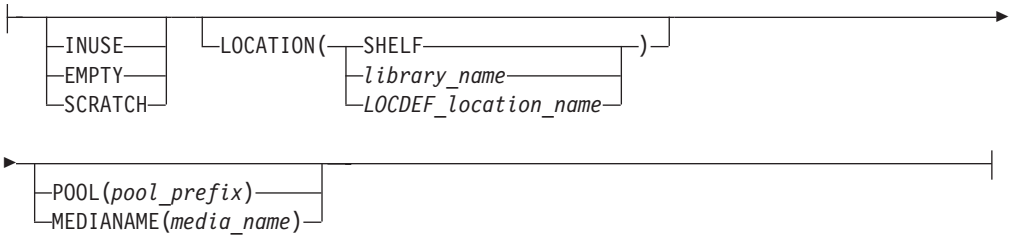
CLIST strings:

SEARCHRACK: Creating a List of Shelf Locations

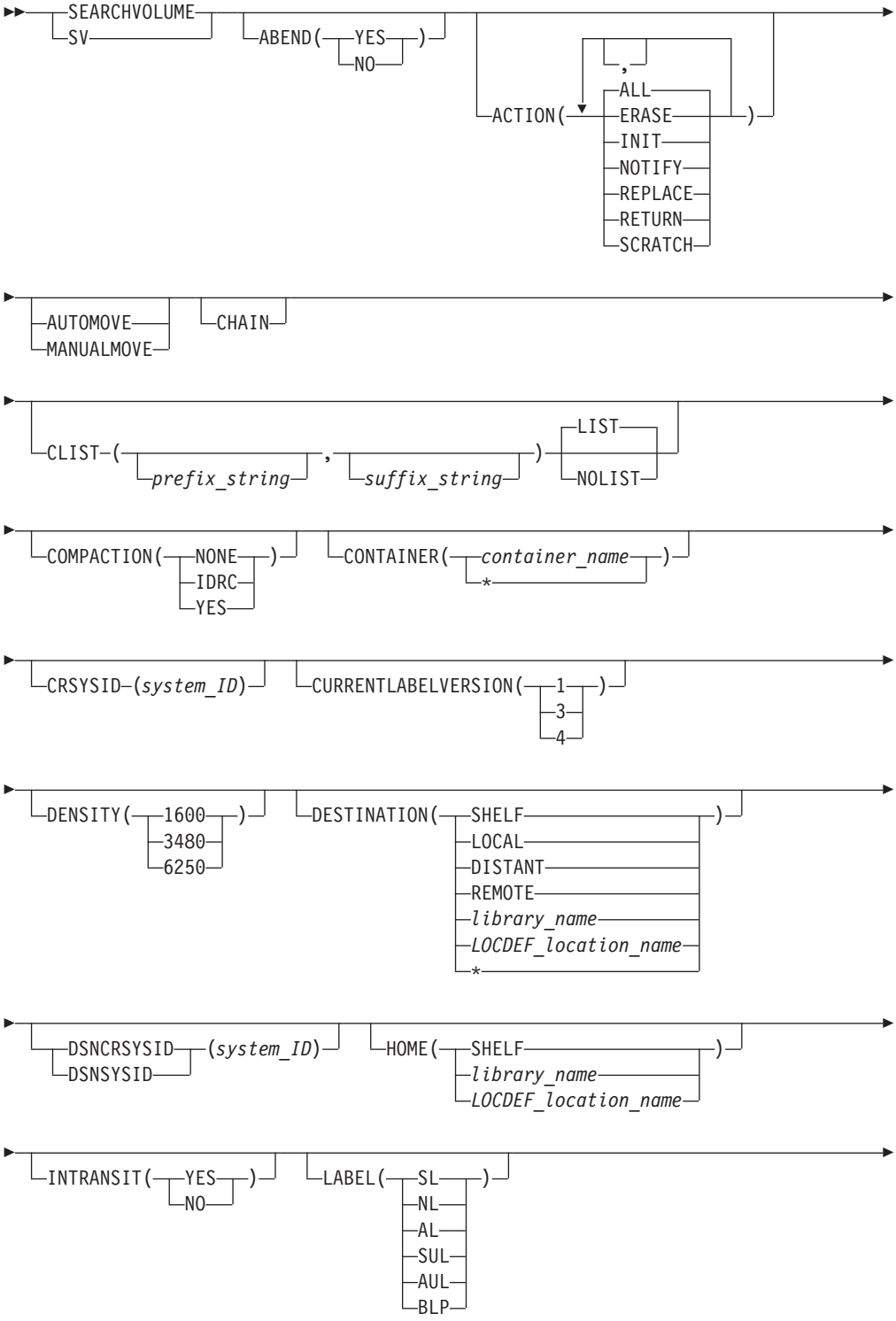


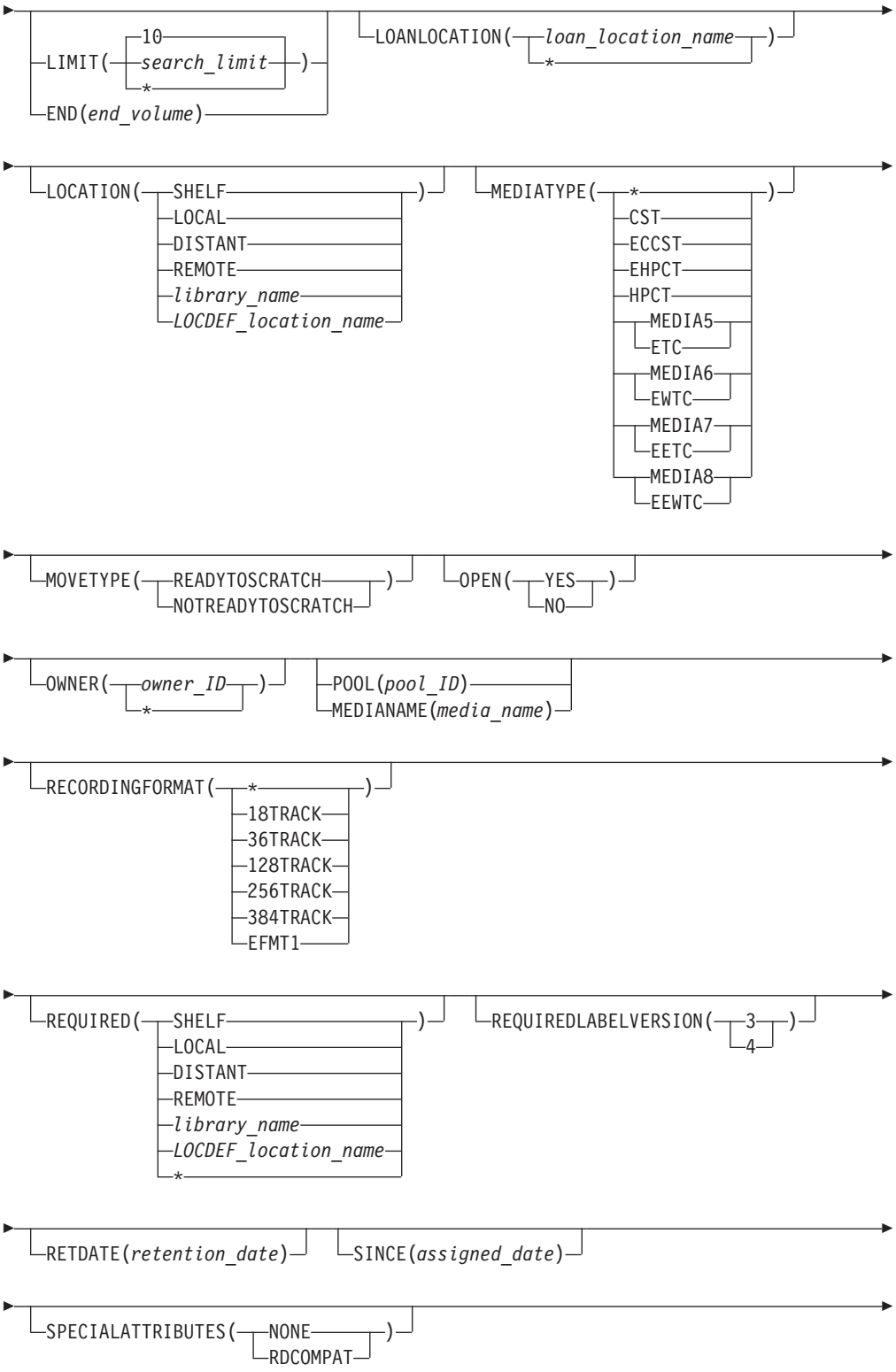
CLIST strings:

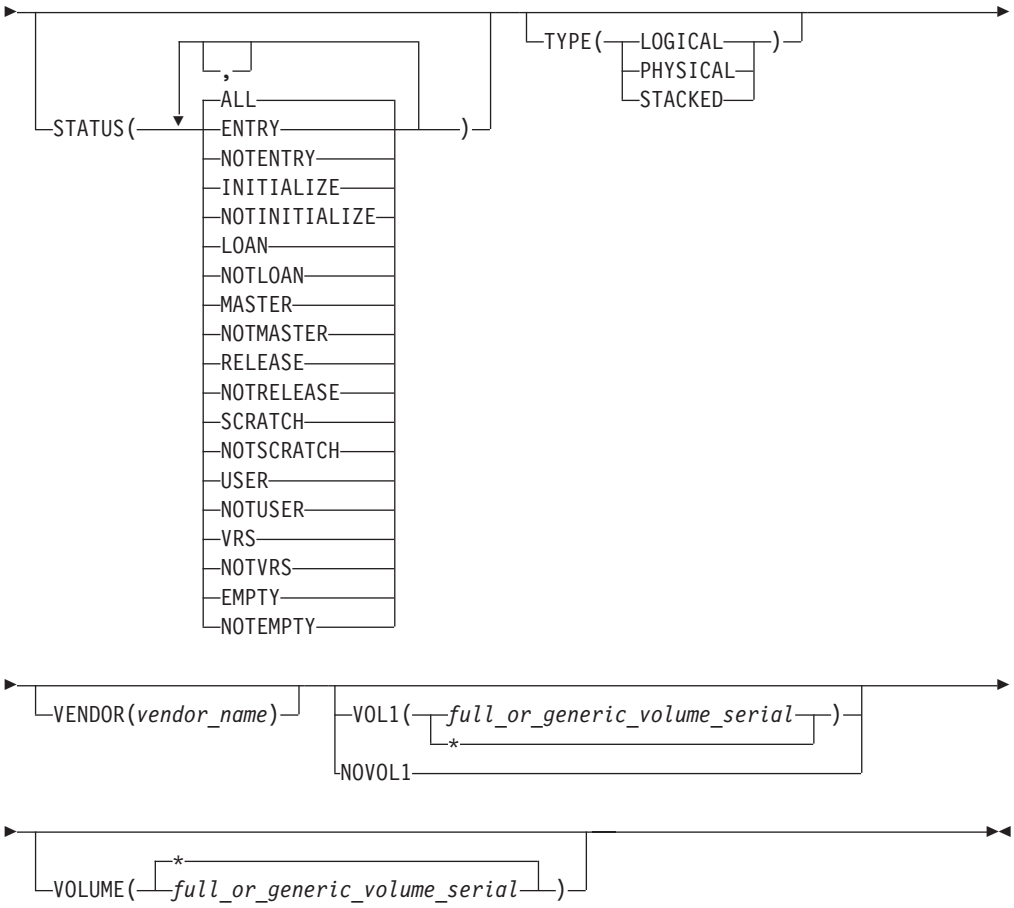
rack number locations:



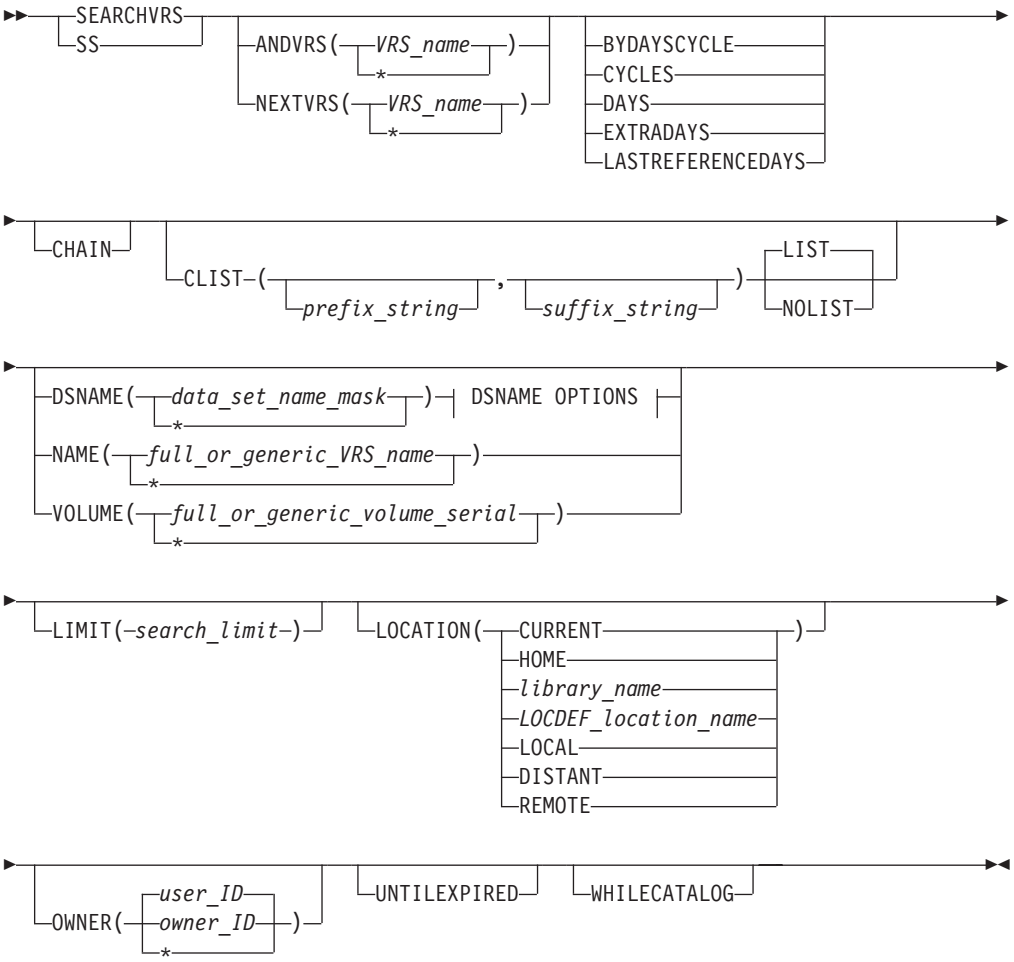
SEARCHVOLUME: Creating a List of Volumes



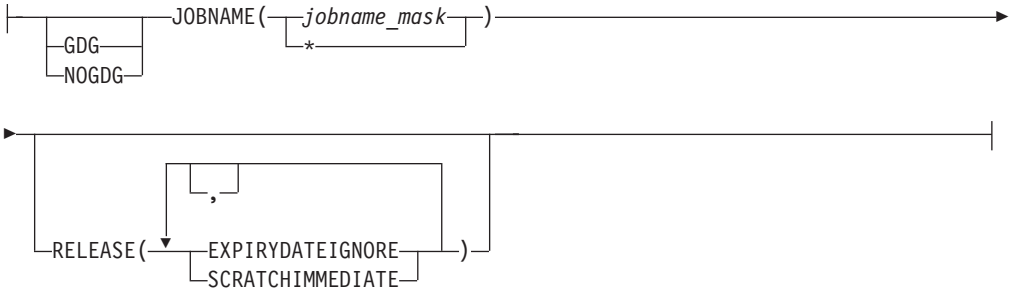

LIMIT(10 search_limit *)
END(end_volume)
LOANLOCATION(loan_location_name *)
LOCATION(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name)
MEDIATYPE(* CST ECCST EHPCT HPCT MEDIA5 ETC MEDIA6 EWTC MEDIA7 EETC MEDIA8 EEWTC)
MOVETYPE(READYTOSCRATCH NOTREADYTOSCRATCH)
OPEN(YES NO)
OWNER(owner_ID *)
POOL(pool_ID)
MEDIANAME(media_name)
RECORDINGFORMAT(* 18TRACK 36TRACK 128TRACK 256TRACK 384TRACK EFMT1)
REQUIRED(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name *)
REQUIREDLABELVERSION(3 4)
RETDATE(retention_date)
SINCE(assigned_date)
SPECIALATTRIBUTES(NONE RDCOMPAT)


STATUS(
,
ALL
ENTRY
NOTENTRY
INITIALIZE
NOTINITIALIZE
LOAN
NOTLOAN
MASTER
NOTMASTER
RELEASE
NOTRELEASE
SCRATCH
NOTSCRATCH
USER
NOTUSER
VRS
NOTVRS
EMPTY
NOTEMPTY
)
TYPE(
LOGICAL
PHYSICAL
STACKED
)
VENDOR(vendor_name)
VOL1(
full_or_generic_volume_serial
*
)
NOVOL1
VOLUME(
*
full_or_generic_volume_serial
)

SEARCHVRS: Creating a List of Vital Record Specifications



DSNAME OPTIONS:



Appendix B. DFSMSrmm ISPF Dialog Fast Path Commands

Table 44 lists the fast path commands you can use within the DFSMSrmm ISPF Dialog.

Table 44. 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.
EJECT Display	Show tape EJECT status.
EJECT BULK	Change tape EJECT entry to BULK.
EJECT CONVENIENCE	Change tape EJECT entry to CONVENIENCE.
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.

© Copyright IBM Corp. 1992, 2004

Table 44. DFSMSrmm ISPF Dialog Fast Path Commands (continued)

Issue The Fast Path Command	To
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.
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.
REPORT	Display the Report Generator Menu.
REUSE Display	Show REUSE status.
REUSE OFF	Do not reuse saved variables to prime the DFSMSrmm ISPF dialog.
REUSE ON	Reuse saved variables to prime the DFSMSrmm ISPF dialog.
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 ADDSTV	Add stacked 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.

Table 44. DFSMSrmm ISPF Dialog Fast Path Commands (continued)

Issue The Fast Path Command	To
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.

Appendix C. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`www.ibm.com/servers/eserver/zseries/zos/bkserv/`

One exception is command syntax that is published in railroad track format; screen-readable copies of z/OS books with that syntax information are separately available in HTML zipped file form upon request to mhvrcfs@us.ibm.com.

© Copyright IBM Corp. 1992, 2004

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

© Copyright IBM Corp. 1992, 2004

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Programming interface information

This publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSMSrmm.

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

IBM
Common User Access
DB2
DFSMSdfp
DFSMShsm
DFSMSrmm
DFSORT
IBMLink
MVS
OS/390
RACF
Service Director
TotalStorage
z/OS
z/VM

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual or view the *Glossary of Computing Terms* located at:

http://www.ibm.com/ibm/terminology/

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

See: This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

A

abend. Abnormal end of task.

AL. American National Standards Label.

AMODE. Addressing mode.

ANDVRS. An RMM ADDVRS TSO subcommand operand. See also *Using AND*.

ANSI. American National Standards Institute.

APAR. Authorized program analysis report.

API. Application Programming interface.

ASA. American Standards Association.

assigned date. The date that the volume is assigned to the current owner. Assigned date is not meaningful for a scratch volume.

AUL. ANSI and user header or trailer label.

automated tape library data server. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. Contrast with *manual tape library*. See also *tape library*.

automatic 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 catalog environment.

BCS. Basic catalog structure.

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*.

BLP. Bypass label processing.

BTLS. Basic Tape Library Support.

built-in storage location. One of the Removable Media Manager defined storage locations: LOCAL, DISTANT, and REMOTE.

© Copyright IBM Corp. 1992, 2004

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 Total Storage Enterprise Automated Tape Library (3494), IBM TotalStorage Enterprise Automated Tape Library (3495), or a manual tape library, 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, the act of logically removing a tape cartridge from the manual tape library by deleting or updating the tape volume record in the tape configuration database.

cartridge entry. For either an IBM Total Storage Enterprise Automated Tape Library (3494), IBM TotalStorage Enterprise Automated Tape Library (3495), or a manual tape library, 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 scratch or private category 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.

client. (1) A user. (2) A consumer of resources or services. (3) A functional unit that receives shared services from a server. (4) A system that is dependent on a server to provide it with programs or access to programs. (5) On a network, the computer requesting services or data from another computer.

client-server. (1) In TCP/IP, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and waits for a response. The requesting program is called a client; the answering program is called a server. (2) A model of computer interaction in which a server provides resources for other systems on a network, and a client accesses those resources. See also *client, client-server relationship, server*.

client-server relationship. Any process that provides resources to other processes on a network is a *server*. Any process that employs these resources is a *client*. A machine can run client and server processes at the same time.

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 TotalStorage Enterprise Automated Tape Library (3494) and the IBM TotalStorage Enterprise Automated Tape Library (3495) 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 TotalStorage Enterprise Automated Tape Libraries (3494) only.

convenience output. The process of removing a small number of tape cartridges from the IBM TotalStorage Enterprise Automated Tape Library (3494) or the IBM TotalStorage Enterprise Automated Tape Library (3495) 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 data set is the date that the data set 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. Direct access storage device.

DASD fast write. An extended function of some models of the IBM 3990 Storage Control in which data is written concurrently to cache and 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 control block (DCB). A control block used by access method routines in storing and retrieving data.

data entry panel. A panel in which the user communicates with the system by filling in one or more fields.

Data Facility Storage Management Subsystem (DFSMS). An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

Data Facility Sort. 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.

DCB. Data control block.

device. This term is used interchangeably with unit. You mount a tape on a unit or device, such as a 3490.

DFSMSdfp. A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMShsm-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, DFSMSrmm-managed volume*.

DFSMShsm-owned volume. A storage volume on which DFSMShsm stores backup versions, dump copies, or migrated data sets.

DFSMSrmm. A DFSMS functional component or base element of z/OS, 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, 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*.

DNS. Domain Name System.

Domain Name System. In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

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

EHPCT. Extended High Performance Cartridge Tape.

eject. The process used to remove a tape volume from a system-managed library. For an automated tape library dataserver, the volume is removed from its cell location and moved to the output station. For a manual tape library, the volume is not moved, but the tape configuration database is updated to show that the volume no longer resides in the manual tape library.

empty bin. A bin that can accept a volume.

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*.

EREP. Environmental Record Editing and Printing program.

expanded output. The output produced by the DFSMSrmm application programming interface when you specify OUTPUT=FIELDS and EXPAND=YES. For those subcommands for which expanded output applies, your application program receives more variable data than for standard output.

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.

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.

export list volume. A virtual tape server logical volume containing the list of logical volumes to export.

extended bin support. Enhanced options for managing shelf locations in a storage location including optimized use of the number of bins.

extended extract data set file. A data set created using the DFSMSrmm EDGJRPT exec. The records within the data set combine data set and volume information into single records.

extended record. A record in the DFSMSrmm extract data set that is mapped by the EDGXREXT mapping macro. The record contains both data set and volume information.

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.

extract data set record. A record in an extract data set that is mapped by a DFSMSrmm mapping macro.

F

field format. Field format is where the output consists of Structured Field Introducers and variable data rather than output in line format.

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.

FIPS. Federal Information Processing Standard.

FMID. Function modification identifier.

FRR. Functional recovery routines.

G

generation data group (GDG). A collection of data sets kept in chronological order. Each data set is a generation data set.

generation data set (GDS). One generation of a generation data group.

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.

GDG. Generation data group.

GDS. Generation data set.

giga (G). The information-industry meaning depends upon the context:

1. G = 1 073 741 824(2^{30}) for real and virtual storage.
2. G = 1 000 000 000 for disk storage capacity (for example, a 4 GB fixed disk).
3. G = 1 000 000 000 for transmission rates.

global resource serialization (GRS). A component of z/OS used for serializing use of system resources and for converting hardware reserves on DASD volumes to data set enqueues.

GPR. General purpose register.

GRS. Global resource serialization.

grouping. When creating a report, grouping sorts report output contents into separate groups (and separate pages) based upon field contents.

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.

HCD. Hardware configuration definition.

high-capacity input station. A transfer station, used by the operator to add tape cartridges to the IBM TotalStorage Enterprise Automated Tape Library (3494) or the IBM TotalStorage Enterprise Automated Tape Library (3495), 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.

HPCT. High Performance Cartridge Tape.

I

ICETOOL. The DFSORT multipurpose data processing and reporting utility.

ID. Identifier.

IDRC. Improved data recording capability.

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.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS that allows users and storage administrators access to the storage management functions.

Interactive Problem Control System (IPCS). A system facility that allows interactive problem analysis.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run interactive, panel-driven dialogs.

internal label. The internal label for standard label tapes is recorded in the VOL1 header label, magnetically recorded on the tape media.

Internet Protocol (IP). The TCP/IP layer between the higher-level host-to-host protocol and the local network protocols. IP uses local area network protocols to carry packets in the form of diagrams to the next gateway or destination host.

in transit. A volume state where a volume 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.

inuse bin. A bin that is occupied by a volume and into which no volume can be assigned.

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*.

IP address. The unique 32-bit address that specifies the location of each device or workstation in the Internet. For example, 9.67.97.103 is an IP address.

IPCS. Interactive Problem Control System.

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.

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.

kilo (K). The information-industry meaning depends upon the context:

1. K = 1024(2^{10}) for real and virtual storage.
2. K = 1000 for disk storage capacity (for example, a 4 KB fixed disk).
3. K = 1000 for transmission rates.

L

Library Control System. The Object Access Method component that controls optical and tape library operations and maintains configuration information.

line format. Line format is where text and variable data are formatted into lines suitable for displaying at a terminal or printing as printed documentation.

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. Logical volumes have a many-to-one association with physical tape media and are used indirectly by MVS applications. They reside in a Virtual Tape Server or on exported stacked volumes. Applications can access the data on these volumes only when they reside in a Virtual Tape Server which makes the data available via its tape volume cache or after the data has been copied to a physical volume through the use of special utilities.

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.

LSR. Local shared resource.

M

management class. (1) A named collection of management attributes describing the retention and backup characteristics for a group of data sets, or for a group of objects in an object storage hierarchy. For objects, the described characteristics also include class transition. (2) In DFSMSrmm, if assigned by ACS routine to system-managed tape volumes, management class 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. 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. An installation-defined set of stand-alone tape drives and the set of tape volumes that can be mounted on those drives.

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. 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 media name are: SQUARE, ROUND, CARTRDGE, 3480.

media type. A value that specifies the volume's media type. Media type can be specified as *, CST, ECCST, HPCT, or EHPCT.

MEDIA 1. Cartridge system tape.

MEDIA 2. Enhanced capacity cartridge system tape.

MEDIA 3. High performance cartridge tape.

MEDIA 4. Extended high performance cartridge tape

MEDIA 5. IBM TotalStorage Enterprise Tape Cartridge.

MEDIA 6. IBM TotalStorage Enterprise WORM Tape Cartridge.

MEDIA 7. IBM TotalStorage Enterprise Economy Tape Cartridge.

MEDIA 8. IBM TotalStorage Enterprise Economy WORM Tape Cartridge.

mega (M). The information-industry meaning depends upon the context:

1. M = 1 048 576(2^{20}) for real and virtual storage.
2. M = 1 000 000 for disk storage capacity (for example, a 4 MB fixed disk).
3. M = 1 000 000 for transmission rates.

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled. See also *migration level 1, migration level 2*.

migration level 1. DFSMShsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage, migration level 2*.

migration level 2. DFSMShsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage, migration level 1*.

moving-in volume. A volume for which a move into a bin has been started, but not yet confirmed.

moving-out volume. A volume for which a move out of a bin has been started, but not yet confirmed.

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 also *Using Next*.

NL. No label.

nonscratch volume. A volume that is not scratch, which means it has valid or unexpired data on it. Contrast with *scratch*.

NSL. Nonstandard label.

O

OAM. Object access method.

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.

OPC/ESA. Operations Planning and Control/Enterprise Systems Architecture.

optical volume. Storage space on an optical disk, identified by a volume label. See also *volume*.

optical disk. A disk that uses laser technology for data storage and retrieval.

option line. 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.

PDS. Partitioned data set.

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*.

PF. Program function key.

physical stacked volume. See *stacked volume*.

physical volume. A volume that has a one-to-one association with physical tape media and which is used directly by MVS applications. It may reside in an automated tape library dataserver or be kept on shelf storage either at vault sites or within the data center where it can be mounted on stand-alone tape drives.

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.

pool ID. The identifier for a pool. You define pool IDs in parmlib member EDGRMMxx.

pooling. The process of arranging shelf locations in the removable media library into logical groups.

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*.

port. (1) An access point for data entry or exit. (2) A receptacle on a device to which a cable for another device is attached.

primary space allocation. Amount of space requested by a user for a data set when it is created. Contrast with *secondary space allocation*.

primary storage. A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1, migration level 2*.

primary 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.

private tape volume. A volume assigned to specific individuals or functions.

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.

PSW. Program status word.

PTF. Program temporary fix.

pull list. A list of scratch volumes to be pulled from the library for use.

PUT. Program update tape.

R

RACF. Resource Access Control Facility.

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 dataserver. The rack number identifies the pool, the external volume serial, and shelf location number for a volume not residing in an automated tape library dataserver. 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.

recording format. For a tape volume, the format of the data on the tape; for example, 18 tracks or 36 tracks.

record-only mode. The operating mode where DFSMSrmm records information about volumes as you use them, but does not validate or reject volumes.

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 start generation zero is the latest generation of a tape. Relative start generation -1 is the previous generation of that tape. Relative start generation -2 is the generation before the previous one.

REMOTE. A DFSMSrmm built-in storage location ID. See also *built-in storage location*.

removable media. See also *volume*.

removable media library. The volumes that are available for immediate use, and the shelves where they could reside.

report. Data that has been selected and extracted according to the reporting tool, the type of report desired, and the formatting criteria.

reporting tool. A REXX exec that builds control statements to enable you to create reports using a reporting utility.

report type. A data source and how it is mapped.

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.

Resource Group. A collection of structured fields that describe the attributes of a resource such as a volume.

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.

REXX. Restructured Extended Executor Language.

RMF. Resource Measurement Facility.

RMM complex (RMMplex). One or more MVS images that share a common DFSMSrmm control data set.

RMODE. Residence mode.

S

SAF. System Authorization Facility.

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*.

SDB. Structured database.

SDSF. Spool display and search facility.

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.

server. (1) A functional unit that provides shared services to workstations over a network; for example, a file server, a print server, a mail server. (2) On a network, the computer that contains the data or provides the facilities to be accessed by other computers in the network. (3) A program that handles protocol, queuing, routing, and other tasks necessary for data transfer between devices in a computer system.

SFI. Structured field introducer.

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*.

SL. Standard label.

slot. See *shelf location*.

SMF. System management facility.

SMP/E. System Modification Program Extended.

SSI. Subsystem interface.

stacked volume. A volume that has a one-to-one association with physical tape media and which is used in a virtual tape server to store logical volumes. A stacked volume is not used by MVS applications but by the virtual tape server and its associated utilities. It may be removed from a virtual tape server to allow transportation of logical volumes to a vault or to another virtual tape server.

standard label. An IBM standard tape label.

standard output. The output produced by the DFSMSrmm application programming interface when you specify OUTPUT=LINES or EXPAND=NO with OUTPUT=FIELDS.

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, the portion of a striped data set, such as an extended format data set, that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

structured field. Output from the DFSMSrmm application programming interface consisting of a Structured Field Introducer and output data.

structured field introducer (SFI). An 8-byte entity that either introduces the beginning of a group of data or introduces output data that immediately follows the introducer.

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.

SUL. IBM standard and user header or trailer label.

SVC. Supervisor call.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications. See also *system-managed storage environment*.

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, this function is provided by DFSMS, DFSORT, and RACF. See also *system-managed storage*.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also *tape library*.

system-managed volume. A DASD, optical, or tape volume that belongs to a storage group. Contrast with *DFSMShsm-managed volume, DFSMSrmm-managed volume*.

system 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 (TCDB). 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, automated tape library data server*.

Tape Library Control System (TLCS). IBM program offering 5785-EAW. DFSMSrmm replaces TLCS.

tape library dataserver. A hardware device that maintains the tape inventory that is associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, and storage of tapes. An automated tape library dataserver that supports system-managed storage of tape volumes. The IBM automated tape library dataservers include the IBM 3494 Tape Library Dataserver and the IBM 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. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

TCDB. Tape configuration database.

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*.

tera (T). The information-industry meaning depends upon the context:

1. T = 1 099 511 627 776(2^{40}) for real and virtual storage.
2. T = 1 000 000 000 000 for disk storage capacity (for example, 4 TB of DASD storage).
3. T = 1 000 000 000 000 for transmission rates.

TLCS. Tape Library Control System.

Transmission Control Protocol (TCP). A stream communication protocol that includes error recovery and flow control.

Transmission Control Protocol/Internet Protocol (TCP/IP). The two fundamental protocols of the Internet protocol suite. The abbreviation TCP/IP is frequently used to refer to this protocol suite. TCP/IP provides for the reliable transfer of data, while IP transmits the data through the network in the form of datagrams. Users can send mail, transfer files across the network, or execute commands on other systems.

TSO. Time Sharing Option.

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 DAD 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 export. A method of exporting a volume by marking a volume as exported by using the DFSMSrmm subcommands.

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 IBM TotalStorage Enterprise Automated Tape Library (3494) or the IBM TotalStorage Enterprise Automated Tape Library (3495), 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 records. A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement. Compare with *disaster recovery*.

vital record specification. Policies defined to manage the retention and movement of data sets and volumes 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.

VOLSER. Volume serial number.

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). (1) 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. (2) In DFSMSrmm, volume serial numbers do not have to match rack numbers.

VTS. Virtual tape server.

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.

world-wide identifier (WWID). Often used in z/OS software as the abbreviation for the world-wide unique cartridge identifier (WWCID). See also *world-wide unique cartridge identifier*

world-wide unique cartridge identifier (WWCID). A permanent identifier associated with a specific tape cartridge, typically stored on the tape itself and the non-volatile cartridge memory.

Write Once, Read Many (WORM). A technology to allow data to be written once to storage media. After that, data is permanent and cannot be altered, but can be read any number of times.

write-to-operator (WTO). An optional user-coded service that allows a message to be written to the system console operator informing the operator of errors and unusual system conditions that may need to be corrected.

WTO. Write-to-operator.

WWCID. See world-wide unique cartridge identifier.

WWID. See world-wide identifier.

Index

Special characters

% (percent sign) 42
&asterisk; (asterisk) 42

A

ABEND
  reserved data set name mask 43
  resetting the flag 224
  searching for data sets marked open 316, 332, 333
  setting the flag 221
  volume information
    non-scratch volume 27, 186
    scratch volume 28, 186
ABEND vital record specification 57
accessibility 421
accounting information 290
Action bar, how used 9
active requests, number of 133
ADDBIN Subcommand
  overview 166
  return codes 167
  syntax 166
ADDDATASET Subcommand
  overview 169
  return codes 175
  syntax 169
adding
  a creating job name 172
  an exported logical volume 192
  bin numbers 24, 166
  bin numbers to a built-in storage location 183
  bin numbers to an installation defined location 183
  compaction information 191, 245
  data set information 35, 169
  exported logical volumes 192
  ISO/ANSI label for VOL1 200
  ISO/ANSI label version 193, 194
  logical volumes 201
  media type information 196
  mediatype information 252
  owner ID 31, 176
  owner information 31, 176
  physical volumes 192, 201
  rack number 24, 182
  recording formation information 198, 257
  scratch volumes 28
  shelf location information 166, 182
  software product information 33, 179
  vital record specification 51, 205
  volumes 186
  volumes that reside in a manual tape library 26
adding duplicate volume serial numbers 28
ADDOWNER Subcommand
  overview 176
  return codes 178
  syntax 176
ADDPRODUCT Subcommand
  overview 179
  return codes 180
  syntax 179
ADDRACK Subcommand
  overview 182
  return codes 185
  syntax 183
ADDVOLUME Subcommand
  overview 186
  return codes 204
  syntax 187
ADDVRS example 96
ADDVRS Subcommand
  overview 205
  return codes 218
  syntax 206, 207, 209
ADMIN fast path command 417
ADMIN operand 3
American date format 7
applying movement and retention policies 43
archiving data 73
assigning
  a new volume serial number 253
  access to volumes 190
  scratch volume 16, 281
authorizing users
  STGADMIN.EDG.IGNORE.TAPE 29
  using STGADMIN.EDG.IGNORE.TAPE 29
  using STGADMIN.EDG.LISTCONTROL 113
  using STGADMIN.EDG.MASTER 29
automated tape library
  cartridge entry processing 76
  confirming moves 79
  convenience output station 77
  ejecting volumes from 77, 275
  high capacity station 77
automatic
  data set information recording 35
  deletion of data set information 36
  owner notification 32, 105
  processing 102
  recovery 147
  release action 103, 104
  releasing of volumes 102
  return to scratch status 103
  volume information recording 25
  volume release 102
AUTOMOVE operand 237

B

background processing 8
batch job
  submitting 12, 165
batch processing 151
bin number
  adding 24, 166

© Copyright IBM Corp. 1992, 2004

bin number *(continued)*
defining 23, 166
deleting 25, 263
details 111, 285
initial 24, 25
line operators 123, 124
list 115, 310
building an export/import list 343
built-in storage locations
adding bin numbers to 166
deleting bin numbers from 263
displaying bin number details 285
bypass label processing
conditions when not permitted 143

C

calculating
volume retention date 65
volume usage value 306
canceling outstanding volume moves 238, 252
cartridge entry processing 76
catalog date format 66
catalog retention period 57, 216
chaining
data sets in a multi-volume data set 333
data sets in the same multi-volume data set 316
chaining vital record specifications 39
CHANGEDATASET Subcommand
overview 219
return codes 228
syntax 219
CHANGEOWNER Subcommand
overview 229
return codes 231
syntax 229
CHANGEPRODUCT Subcommand
overview 232
return codes 233
syntax 232
CHANGEVOLUME Subcommand
overview 234
return codes 262
syntax 239
changing
data set information 219
data set information automatically 21
data set information with RMM CHANGEDATASET subcommand 36
data set name 224
expiration date 18, 107
home location with RMM CHANGEVOLUME subcommand 236
label version 248
list sort order 117
location name 251
owner information 19, 32
owner information with RMM CHANGEOWNER subcommand 229
pending status or scratch status 236
pool ID 77
changing *(continued)*
release date 18, 106
retention period 18
shelf location 77
software product information 34, 232
vital record specification 52
volume information 29, 234
volume release date 18, 106
volume serial number 253
volume type 259
changing information for multiple volumes 30, 31
character set
chart xxii
use in statement xxii
checking dfsmsrmm status 133
cleaning up name vital record specifications 89
clearing loan location 251
clearing the loan location 79
clearing volume location information 247
CLIST
confirming volume moves 83
confirming volume release actions 109
creating a 128
ejecting volumes 77
closed-cycle GDGs 69
command
abbreviations 163, 164
ADDBIN, TSO subcommand 166
ADDDATASET, TSO subcommand 169
ADDOWNER, TSO subcommand 176
ADDPRODUCT, TSO subcommand 179
ADDRACK, TSO subcommand 182
ADDVOLUME, TSO subcommand 186
ADDVRS, TSO subcommand 205
batch job 165
CHANGEDATASET, TSO subcommand 219
CHANGEOWNER, TSO subcommand 229
CHANGEPRODUCT, TSO subcommand 232
CHANGEVOLUME, TSO subcommand 234
CONFIRM 8
DATE 7
DELETEBIN, TSO subcommand 263
DELETEDATASET, TSO subcommand 266
DELETEOWNER, TSO subcommand 268
DELETEPRODUCT, TSO subcommand 270
DELETERACK, TSO subcommand 272
DELETEVOLUME, TSO subcommand 275
DELETEVRS, TSO subcommand 279
DFSMSrmm 163
END 11
fast path 10
GETVOLUME, TSO subcommand 281
LISTBIN, TSO subcommand 285
LISTCONTROL, TSO subcommand 287
LISTDATASET, TSO subcommand 295
LISTOWNER, TSO subcommand 298
LISTPRODUCT, TSO subcommand 300
LISTRACK, TSO subcommand 302
LISTVOLUME, TSO subcommand 304
LISTVRS, TSO subcommand 307
PRINT-HI, ISPF command 15, 129

command *(continued)*
PRINT, ISPF command 15, 129
processing 8
RETURN 11
SEARCHBIN, TSO subcommand 310
SEARCHDATASET, TSO subcommand 315
SEARCHPRODUCT, TSO subcommand 322
SEARCHRACK, TSO subcommand 325
SEARCHVOLUME, TSO subcommand 329
SEARCHVRS, TSO subcommand 347
SORT 120
specifying 163
TSO subcommand 163
Command Menu 4
COMMAND operand 3
COMMANDS fast path command 417
compaction information
adding 191, 245
CONFIRM command 8
confirm delete option 8
confirming
all pending moves 55, 236
delete and release requests 8
deletions
data set information 37
software product 35
line operators 108
list 126
manual release action 104, 107
using CHANGEVOLUME 79, 236
volume move 79, 236
volume release 110
console messages 132
container
defining 192
CONTROL
fast path command 417
operand 3
control data set
calculating volume usage 306
control record information 287
defining resources 23
displaying details 287
system options information 113
controlling data set recording 26
controlling tape erasure 154
controlling tape initialization 154
converting from closed-cycle gdgs to scratch pools 69
correcting data set name 224
correcting volume serial number 253
creating
a CLIST 128
bin number list 115
data set list 115
dropship list 126
list of resources 115
rack number list 115
REXX exec 363
shelf location list 115
software product list 115
vital record specifications list 115
creating *(continued)*
volume list 115
volume movements and actions list 127
creating a volume export list 345
creating job name
changing with RMM CHANGEDATASET subcommand 228
default used by DFSMSrmm 223
definition 58
displaying with RMM LISTDATASET subcommand 295
specifying with RMM ADDDATASET subcommand 172
cycles date format 66

D

data class recorded by DFSMSrmm 35
data set
adding information 35, 169
automatic recording of information 35
chaining data sets in a multi-volume data set 333
chaining data sets in the same multi-volume data set 316
changing information 36, 219
changing information automatically 21
changing name 224
defining 35, 169
deleting information 36, 266
details 111, 295
displaying all the data sets in a multi-volume set 21
fully qualified name 41
generic data set name 42
how to define data set name 41
line operators 123, 124
list 115, 315
management class 295
marking incorrect 221
name mask 212
non-restricted operands or fields 219
restricted operands or fields 219
retain by job name 43
retaining based on catalog status 56
retaining by job name 58
retaining until expired 56, 57
retaining while cataloged 56
retaining while open 57
retention date 65
saved commands 11
searching for data sets in a multi-volume data set 316, 333
searching for data sets marked open 316, 332
sequence number 36, 112
uncataloging 37
updating information 36, 219
updating information automatically 21
vital record specification management value 295
data set recording 26
data set vital record specification
adding 51
definition 39

data set vital record specification *(continued)*
matching data set names 43, 44
specifying data set names 41
DATASET
fast path command 417
operand 3
DATE command 7
date format
catalog date 66
cycles date 66
setting for a session 7
used in initialization messages 137
defining release policies 62
defining vital record specification with RMM ADDVRS subcommand
a creating job name 172
a scratch volume 26
data set 35
exported logical volumes 192
home location 70
library names 71
loan location 78
logical volumes 201
master volume 27
movement policies 70
original expiration date 106
owner 31, 176
physical volumes 201
resource 23
retention policies 55
scratch volume 28
shelf location 23, 166, 182
software product 33, 179
user volume 27
vital record specification 51
volume 25, 186
DELETEBIN Subcommand
overview 263
return codes 265
syntax 263
DELETEDATASET Subcommand
overview 266
return codes 267
syntax 266
DELETEOWNER Subcommand
overview 268
return codes 268
syntax 268
DELETEPRODUCT Subcommand
overview 270
return codes 271
syntax 270
DELETERACK Subcommand
overview 272
return codes 274
syntax 272
DELETEVOLUME Subcommand
overview 275
return codes 278
syntax 276
DELETEVRS Subcommand
overview 279
return codes 280
syntax 279
deleting
an empty rack with a medianame that does not match VLPOOL 273
bin number 25, 263
bin numbers from a built-in storage location 272
bin numbers from an installation defined location 272
confirm
data set information 37
software product information 35
data set information 36, 266
owner ID 32, 268
owner information 32, 268
rack number 25, 272
shelf location information 25, 272
software product information 34, 270
vital record specification 52, 279
volume information 31, 275
delimiters xxii
DFSMSdfp labeling support 194, 250
DFSMShsm
backup and migration tapes 217
migration tapes, moving 73
retaining DFSMShsm ABARS backups 217
DFSMSrmm
getting started 1
ISPF dialog 1
navigating through the DFSMSrmm dialog 9
reason codes 355
refreshing the DFSMSrmm installation exits 133
removing DFSMSrmm from the system 135
resources 23
return codes 355
starting 1
stopping 11
tape initialization and erase control 154
TSO subcommands 163
wrong label processing 154
DFSMSrmm ISPF exec
operands 3
using 2
diagnosing errors 3, 103
dialog
creating CLISTS of executable commands 128
exiting 11
fast path commands 10
options 7
starting 1
disability 421
disaster recovery 55
displaying
all the data sets in a multi-volume set 21
bin number details 285, 302
control record information 287
data set details 295
DFSMSrmm control data set information 113
disposition control options 113

displaying *(continued)*
installation options 113
owner details 298
parmlib options information 287
rack number details 302
shelf location details 285, 302
software product details 300
vital record specification details 307
volume details 304
displaying required destination 338
disposition control
default sticky label description 160
displaying options 113
sticky label support 160
system options information 290
documents, licensed xviii
dropship list 126
dummy tag
ABEND 57
matching to data set names 43, 44
OPEN 57
retaining 55
duplicate volume serial number 29
adding 28
adding with RMM ADDVOLUME subcommand 203
ignoring with EDGUX100 29
managing 33
redefining 29
searching for duplicate volumes 341

E

EDG0001I 137, 146
EDG0101I 138
EDG0103D 138
EDG0104E 138
EDG0105I 138
EDG0110D 137
EDG0114I 138
EDG0115D 138
EDG0123D 138
EDG019VM installation exit 142
EDG0204I 137
EDG0215D 138
EDG0228E 139
EDG0300I 148
EDG0301I 148
EDG0302I 148
EDG0303D 148
EDG0304I 148
EDG0305I 148
EDG0353I 139
EDG0358D 139
EDG0361D 139
EDG1001A 138
EDG1101I 139
EDG1105I 139
EDG1106I 139
EDG1107D 140
EDG2103D 146
EDG2104E 146
EDG2107E 147
EDG2108E 147
EDG2110I 147
EDG2111I 147
EDG2112I 147
EDG2113I 147
EDG2114I 147
EDG2115I 147
EDG2116A 147
EDG4000D 145
EDG4001D 145
EDG4005E 146
EDG4006E 146
EDG4007E 145
EDG4008A 145
EDG4010D 145
EDG4012D 146
EDG4013I 141
EDG4021I 143
EDG4023I 143
EDG4024I 143
EDG4025I 143
EDG4026I 143
EDG4027I 144
EDG4028I 144
EDG4032I 144
EDG4033I 144
EDG4035I 144
EDG4036I 144
EDG4041I 144
EDG4048I 145
EDG4049I 145
EDG6627A 141
EDG6628A 141
EDG6663D 141
EDG8121D 148
EDG8122D 148
EDG8123D 149
EDG9115I 140, 147
EDG9116I 147
EDGINERS utility
automatic processing 104
manual processing 149
using EDGINERS 149
wrong label processing 154
EDGLABEL procedure
description 149
message responses 155
EDGRPTD utility 111
EDGRVCLN REXX EXEC 89
EDGUX100 installation exit 29
EDGV01D 142
ejecting
cartridges to an output station 8
volumes by building a list of DFSMSrmm subcommands 77
volumes from tape libraries 76
volumes with ISMF or MVS console command 262
volumes with RMM DELETEVOLUME subcommand 275
empty rack or bin number 25

END command 11
ending a search for volume information 334
ERASE request format 157
erase volume release action 19, 104
erasing volumes 104, 149
error diagnosis 3, 103
European date format 7
examples
  adding
    a data set with RMM ADDVOLUME subcommand 175
    a duplicate volume with RMM ADDVOLUME subcommand 203
    a new owner with RMM ADDOWNER subcommand 178
    a new product with RMM ADDPRODUCT subcommand 180
    empty bin to the storage location with RMM ADDBIN subcommand 167
    empty bins to a storage location with RMM ADDBIN subcommand 167
    empty bins to a storage location with RMM ADDRACK subcommand 184
    empty racks to the removable media library with RMM ADDRACK subcommand 184
    tape cartridge with RMM ADDVOLUME subcommand 203
    volumes in a system-managed library with RMM ADDVOLUME subcommand 204
    volumes using status information from the TCDB 204
    WORM volumesTCDB 204
  associating
    a volume to a software product with RMM CHANGEVOLUME subcommand 261
  building
    a list of eject commands with RMM SEARCHVOLUME subcommand 342
  changing
    owner information with RMM CHANGEOWNER subcommand 230
    owner of a software product with RMM CHANGEPRODUCT subcommand 233
    the MANAGEMENTVALUE assigned to a data set 228
  confirming
    volume actions with RMM CHANGEVOLUME subcommand 261
  creating
    a list of bin numbers with RMM SEARCHBIN subcommand 313
    a list of data sets with RMM SEARCHDATASET subcommand 319
    a list of duplicate volumes with RMM SEARCHVOLUME subcommand 341
    a list of empty or available bin numbers with RMM SEARCHBIN subcommand 312
    a list of scratch volumes with RMM SEARCHRACK subcommand 327
    a list of software products with RMM SEARCHPRODUCT subcommand 324
examples *(continued)*
  creating *(continued)*
    a list of volumes that belong to an owner with RMM SEARCHVOLUME subcommand 342
  defining
    vital record specification with RMM ADDVRS subcommand 217
  deleting
    a data set vital record specification 280
    a volume vital record specification with RMM DELETEVRS subcommand 280
    information about a volume with RMM DELETEVOLUME subcommand 278
    information about data sets with RMM DELETEDATASET subcommand 267
    software product information with RMM DELETEPRODUCT subcommand 270
    user ID and node for an owner with RMM CHANGEOWNER subcommand 230
    user IDs with RMM CHANGEVOLUME subcommand 261
  displaying
    information for a single data set with RMM LISTDATASET subcommand 296
    installation's options and rules with RMM LISTCONTROL subcommand 289
    location definitions with RMM LISTCONTROL subcommand 293
    security class masks with RMM LISTCONTROL subcommand 291
  JCL to run a batch TSO TMP 165
  listing
    vital record specifications with RMM SEARCHVRS subcommand 352
  managing
    special dates with RMM ADDVRS subcommand 217
  marking
    a volume as no longer in transit with RMM CHANGEVOLUME subcommand 262
  moving
    a volume to an automated tape library with RMM CHANGEVOLUME subcommand 261
  releasing
    volumes associated with a software product with RMM DELETEPRODUCT subcommand 270
  removing
    bin numbers with RMM DELETEBIN subcommand 264
    shelf location information with RMM DELETEBIN subcommand 264
    shelf location information with RMM DELETERACK subcommand 274
  requesting
    a volume from the default scratch pool with RMM GETVOLUME subcommand 284
    information a shelf location with RMM LISTBIN subcommand 286
    information a shelf location with RMM LISTRACK subcommand 303

examples *(continued)*
requesting *(continued)*
information about a vital record specification with RMM LISTVRS subcommand 308
owner information with RMM LISTOWNER subcommand 298
software product information with RMM LISTPRODUCT subcommand 300
volume serial information with RMM LISTVOLUME subcommand 305
volume statistics with RMM LISTVOLUME subcommand 305
retaining
DFSMShsm ABARS backups with RMM ADDVRS subcommand 217
DFSMShsm backup and migration tapes with RMM ADDVRS subcommand 217
retention by cycles 41
setting
a creating job name 228
testing
return codes in a CLIST 355
return codes in a REXX procedure 355
transferring
volume ownership with RMM DELETEOWNER subcommand 268
updating
data set information with RMM CHANGEDATASET subcommand 227
using TSO help 164
Exceptions
moving volumes from storage locations 79
exec
DFSMSrmm ISPF 2
REXX 363
Exit Menu 11
exiting the DFSMSrmm dialog 11
expiration
date 18, 107
JCL date 59
processing 102
expiration date protection 59, 106
expirydateignore 59, 62, 107
exported logical volume
adding 192
EXTRADAYS example 96

F

fast path commands
issuing 10
fields, restricted and non-restricted 219, 235
filtering with data set name masks 43
foreground processing 8
freeing up space in system-managed libraries 71
fully qualified data set name 41
function menus 5

G

general user
getting started with 1
generation data group (GDG)
base names 48
closed-cycle 69
converting to use scratch 70
generic data set
description 42
using 212
generic data set name 42
generic job name description 45
getting started 1
GETVOLUME Subcommand
overview 281
return codes 284
syntax 281
global confirmation
examples 82
performing 237
undoing movement and action confirmation 237
grouping volumes 21

H

help
online RMM TSO subcommand help 164
home location
changing 71
changing with CHANGEVOLUME subcommand 236
setting 70, 195

I

I/O errors on a volume 103
IEF233A 141
IFG019VM installation exit 142
ignoring duplicate volume serial numbers 29
ignoring expiration date 59, 102
implementing a data archive process 73
in transit 79
information about your resources 5
information recording performed by DFSMSrmm 35
INIT request format 157
initial
rack or bin number 24, 25
volume serial number 28
initialization messages 137
initialize volume release action 19, 104
initializing volumes 104, 149
installation defined storage locations
adding bin numbers to 166
deleting bin numbers from 263
displaying bin number details 285
obtaining information about 24
installation exits
EDGUX100 29
messages 148
refreshing the DFSMSrmm installation exits 134

inventory management
confirming volume movements 79
report 111
ISO date format 7
ISO/ANSI
changing label version 248
changing the label version 258
defining label version 193, 194
defining label version in the VOL1 label 200
labeling 149
searching for ISO/ANSI label information 334
Version 3 151
Version 4 151

J

JCL expiration date 59
job name
controlling policy selection 43
creating job name definition 58
generic job name 45
how to define 45
retaining data sets by job name 58
using with ADDDATASET subcommand 169
job name mask 45
journal
operator system error messages 146
rexx variable 372
system options information 113, 290
Julian date format 7

K

keyboard 421

L

LABEL procedure
description 149
labeling ISO/ANSI tape labels 149
last referenced retention type 56
LIBRARIAN fast path command 417
LIBRARIAN operand 3
licensed documents xviii
limit, search 115
limiting the search for volume information 334
line operators
confirm 108, 128
data set 123
rack and bin number 123
release volumes 110
shelf location 123
software product 123
using 123
vital record specification 123
volume 123
LISTBIN Subcommand
overview 285
return codes 286
syntax 285
LISTCONTROL Subcommand
overview 287
return codes 294
syntax 287
LISTDATASET Subcommand
overview 295
return codes 297
syntax 295
LISTOWNER Subcommand
overview 298
return codes 299
syntax 298
LISTPRODUCT Subcommand
overview 300
return codes 301
syntax 300
LISTRACK Subcommand
overview 302
return codes 303
syntax 302
lists
bin number 310, 325
changing sort order 117
CLISTS of executable commands 128
confirm volume moves 126
creating 115
data set 115, 315
dropship 126
printing 15, 129
rack number 325
requesting 115
scratch volume 124
search limit 115
shelf location 310, 325
software product 115, 322
sorting 117
using 114
vital record specification 347
volume 115, 329
LISTVOLUME Subcommand
overview 304
return codes 306
syntax 304
LISTVRS Subcommand
overview 307
return codes 309
syntax 307
loan location
changing 259
clearing loan location 251
defining 192
moving to a 78
returning from a 79
LOCAL fast path command 417
LOCAL operand 3
location
changing 251
location definitions 293
LOCDEF command 73
logical volume, changing 259
LookAt message retrieval tool xviii

M

management class
  defining management and retention policy 205
  displaying data set information 295
  recorded by DFSMSrmm 35
  specifying 59, 211
managing
  data sets left open by a system failure 57
  data sets that were closed by ABEND processing 57
  multiple data sets with data set masks 43
  special dates 217
  vital records 55
  volumes with the same volume serial number 28, 29
manual move control 73
manual processing 102
manual tape library
  adding volumes that are resident in 26
  moving volumes to 75
MANUALMOVE operand 237
marking a stacked volume closed 245
marking DFSMSrmm—managed datasets incorrect before restarting jobs 221
mask, data set name 212
masking characters 48
master status volumes, controlling tape initialization 154
master volume
  definition 27
  overwriting 26
  searching for information about 319
matching order for vital record specifications 43, 44
maximum retention period 59
media names 26
media types 26
mediatype information
  adding 196, 252
menu
  Command 4
  Exit 11
  primary option menu 1
  user 4
merging UNTILEXPIRED and WHILECATALOG 61
message retrieval tool, LookAt xviii
messages
  displaying case 290
  EDG0001I 137, 146
  EDG0101I 138
  EDG0103D 138
  EDG0104E 138
  EDG0105I 138
  EDG0110D 137
  EDG0114I 138
  EDG0115D 138
  EDG0123D 138
  EDG0204I 137
  EDG0215D 138
  EDG0228E 139
  EDG0300I 148
  EDG0301I 148
messages *(continued)*
  EDG0302I 148
  EDG0303D 148
  EDG0304I 148
  EDG0305I 148
  EDG0353I 139
  EDG0358D 139
  EDG0361D 139
  EDG1001A 138
  EDG1101I 139
  EDG1105I 139
  EDG1106I 139
  EDG1107D 140
  EDG2103D 146
  EDG2104E 146
  EDG2107E 147
  EDG2108E 147
  EDG2110I 147
  EDG2111I 147
  EDG2112I 147
  EDG2113I 147
  EDG2114I 147
  EDG2115I 147
  EDG2116A 147
  EDG4000D 145
  EDG4001D 145
  EDG4005E 146
  EDG4006E 146
  EDG4007E 145
  EDG4008A 145
  EDG4010D 145
  EDG4012D 146
  EDG4013I 141
  EDG4021I 143
  EDG4023I 143
  EDG4024I 143
  EDG4025I 143
  EDG4026I 143
  EDG4027I 144
  EDG4028I 144
  EDG4032I 144
  EDG4033I 144
  EDG4035I 144
  EDG4036I 144
  EDG4041I 144
  EDG4048I 145
  EDG4049I 145
  EDG6627A 141
  EDG6628A 141
  EDG6663D 141
  EDG8121D 148
  EDG8122D 148
  EDG8123D 149
  EDG9115I 140, 147
  EDG9116I 147
  EDGV01D 142
  IEF233A 141
  initialization 137
  installation exit 148
  operator 136
  system error 146

messages *(continued)*
tape mount 140
tape processing 145
tape validation 142
minimum retention with WHILECATALOG 57, 216
MODIFY command 132
mount message processing 140
mouse button 9
movement policies 70
moving
DFSMShsm migration tapes 73
from storage locations 79
list 127
out of system-managed libraries 71
to loan locations 78
volumes 55, 77
volumes using manual move control 73
multivolume chain
last volume in a multivolume chain 262

N

name filtering rules 43
name vital record specification
adding 51, 206
definition 39
example 41
name, data set 212
navigating through DFSMSrmm
using fast path commands 10
using the DFSMSrmm panels 9
NEXTVRS example 96
no label
DFSMSrmm support 158
obtaining information about 142
non-restricted operands or fields 219, 235
non-restricted volume information 235
notification of volume release 20
notify owner release action 19, 105
notifying owners 32

O

obtaining
information about NL tapes 142
information about your resources 5
parmlib suffix 290, 365
online help 164
OPEN
resetting the closed flag 245
resetting the open flag 255
searching for data sets marked open 337
OPEN vital record specification 57
operands, restricted and non-restricted 219, 235
operator
initialization message responses 137
installation exit message responses 148
LABEL procedure message responses 155
messages 136
procedures 131
sticky label support 160
operator *(continued)*
system error message responses 146
tape mount message responses 140
tape processing message responses 145
tape validation message responses 142
options
confirm delete 8
date format 7
dialog 7
ejecting cartridges 8
reusing variable information 8
OPTIONS fast path command 417
overriding Automatic Move Processing 237
overriding expiration date 20, 59
owner
adding information 31, 176
changing information 19, 32, 229
defining 31, 176
deleting information 32, 268
details 111, 298
ID
adding 31, 176
changing 19, 32, 229
defining 31, 176
deleting 32, 268
details 111, 298
notification 32, 105
requesting volume return 104
updating information 19, 32, 229
OWNER fast path command 417
OWNER operand 3

P

parmlib member EDGRMMxx
suffix 365
parmlib options
displaying details 287
implementing new options 133
pending movement and action
confirming 55, 236
confirming using the DFSMSrmm ISPF Dialog 80
for return to scratch volumes 111
permanent I/O errors 103
permanent retention with 99365, 99366, 1999/365, 1999/366 59
physical volume
changing 259
policy
See also &vrs.
movement 55
retention 55
setting up 20
pool
adding rack number 23, 24
changing information for a volume 77
details 127
moving volumes 77
viewing pool ids 28
pooling requirements 21
preventing a stacked volume from moving 255

primary option menu 1
priming DFSMSrmm ISPF dialog panels 8
PRINT command 15, 129
PRINT-HI command 15, 129
printing lists and details panels 15, 129
private volume
  overwriting 26
  searching for information about 319
problem determination aid
  controlling tracing 136
  initialization messages 140
  system error messages 147
problem diagnosis 3
procedures, operator 131
processing
  background 8
  batch, commands in 12
  foreground 8
  saved commands data set 11
processing NL tapes 142
PRODUCT fast path command 417
PRODUCT operand 3
pseudo-GDG
  definition 48
  specifying 212
pulling scratch volumes 124

Q

quiescing DFSMSrmm 135

R

RACF
  STGADMIN.EDG.IGNORE.TAPE 29
  STGADMIN.EDG.LISTCONTROL 113, 287
  STGADMIN.EDG.MASTER 29
  STGADMIN.EDG.RELEASE 275
RACK fast path command 417
rack number
  adding 24, 182
  adding to a pool 24
  defining 23, 182
  deleting 25, 272
  deleting an empty rack with a medianame that does not match VLPOOL 273
  details 111, 302
  initial 24, 25
  line operators 123, 124
  list 115, 325
  removing 255
RACK operand 3
rack pool
  setting up 23
  viewing information about 24
ready to scratch
  confirming status 247
  movement report 111
  NOTREADYTOSCRATCH operand 246
  READYTOSCRATCH operand 246
  using to confirm volume movement 83, 237
reason codes 356
reclaiming volumes from pending status or scratch status 236
Recommendation
  media type 253
  recording format 199
  to change a policy 85
  year and day format 256
recording
  data set information 21, 35
  volume information 18
recording format information
  adding 198, 257
recording format names 26
redefining duplicate volume serial numbers 29
refreshing DFSMSrmm 133
refreshing the DFSMSrmm installation exits 134
relabeling a volume 105
release action
  confirming 104, 107
  list 127
  setting 102
  setting a 19
release date 18
release notification 20
release processing 102
releasing a volume
  associated with a software product 34
  confirming 110
  DELETEVOLUME subcommand 275
  description 102
  line operators 110
  manually 19, 109
  notifying owner 32, 105
  types of 19
releasing volumes 107
removable media library
  moving volumes 55
  pools 23
  retaining volumes 55
  shelf locations 23
  shelves 23
removing
  a rack number 255
  asingle volume from the end of a multivolume chain. 255
  DFSMSrmm from the system 135
  last volume in a multivolume chain 262
  shelf location information with RMM DELETEBIN subcommand 264
  shelf location information with RMM DELETERACK subcommand 274
replace volume release action 19, 103
replacing
  volumes with duplicate volume serial numbers 33
  volumes with permanent errors 103
report
  extract data set 111
  inventory 111
requesting
  bin number details 111, 285

requesting *(continued)*
bin number list 115, 310
confirm delete option 8
control record information 287
data set details 111, 295
data set list 115, 315
dropship list 126
information 111
list of resources 115
lists 111
owner details 111, 298
parmlib options information 287
pool details 127
rack number details 111, 302
rack number list 115, 325
scratch volume 16, 101, 281
shelf location details 111, 285
shelf location list 115, 310
software product details 111, 300
software product list 115, 322
vital record specification details 111, 307
vital record specification list 115, 347
volume details 111, 304
volume list 115, 329
volume movements and actions list 127
volume release 20
required destination 338
reserved words
ABEND 57
OPEN 57
resetting in transit flag 262
resetting the ABEND flag 253
resource
defining 23
information 111
lists 114
system options 113
restarting DFSMSrmm subsystem 131, 132
restricted operands or fields 219, 235
Restrictions
3590 media 105
adding volumes in a multivolume data set 197
CHANGEVOLUME subcommand 236
DELETEOWNER 32
deleting a data set 36, 266
owner information 32
pseudo-GDG vital record specification 49
retaining
based on catalog status 56
by job name 55, 58
by last referenced date 56
data sets 55
until expired 56, 57
volumes 55
by generic volume serial number 50
by specific volume serial number 50
using GDG name 48
with special dates 43
volumes with special dates 59
when closed during ABEND processing 57
while cataloged 56
retaining *(continued)*
while closed during ABEND processing 57
while open 57
retaining a data set using a vital record specification management value 98
retaining data sets as long as they are cataloged 98
retaining data sets based on expiration date 98
retaining data sets based on the last referenced date 96
retaining generation data group data sets 97
retaining volumes by specific volume serial number 50
retention date
calculating 63
obtaining information 338
retention period 58, 107
retention policies 55
return codes
ADDBIN subcommand 167
ADDDATASET subcommand 175
ADDOWNER subcommand 178
ADDPRODUCT subcommand 180
ADDRACK subcommand 184
ADDVOLUME subcommand 204
ADDVRS subcommand 218
CHANGEDATASET subcommand 228
CHANGEOWNER subcommand 231
CHANGEPRODUCT subcommand 233
CHANGEVOLUME subcommand 262
DELETEBIN subcommand 265
DELETEDATASET subcommand 267
DELETEOWNER subcommand 268
DELETEPRODUCT subcommand 271
DELETERACK subcommand 274
DELETEVOLUME subcommand 278
DELETEVRS subcommand 280
description 355
GETVOLUME subcommand 284
LISTCONTROL subcommand 294
LISTDATASET subcommand 297
LISTOWNER subcommand 299
LISTPRODUCT subcommand 301
LISTRACK subcommand 286, 303
LISTVOLUME subcommand 306
LISTVRS subcommand 309
SEARCHBIN subcommand 313
SEARCHDATASET subcommand 321
SEARCHPRODUCT subcommand 324
SEARCHRACK subcommand 328
SEARCHVOLUME subcommand 346
SEARCHVRS subcommand 353
RETURN command 11
return to owner release action 19, 104
return to scratch release action 19, 103
returning a volume to scratch status 62
reversing movement and action confirmation 237
REXX exec
creating 363
testing return codes 355
using TRACE in 3
variables used 363

S

scratch pool
  converting from closed-cycle gdgs 69
  volume
    adding 28
    assigning 16, 101
    defining 26
    list 124
    requesting 16, 101
scratch volume
  assigning 281
  requesting 281
  searching for information about 319
scratchimmediate 62, 107
search limit 115
SEARCHBIN Subcommand
  overview 310
  return codes 313
  syntax 311
SEARCHDATASET Subcommand
  overview 315
  return codes 321
  syntax 315
searching
  for data sets in the same multi-volume data set 316, 333
  for data sets marked open 316, 332, 337
SEARCHPRODUCT Subcommand
  overview 322
  return codes 324
  syntax 322
SEARCHRACK Subcommand
  overview 325
  return codes 328
  syntax 325
SEARCHVOLUME Subcommand
  overview 329
  return codes 346
  syntax 330
SEARCHVRS Subcommand
  overview 347
  return codes 353
  syntax 347
security class masks 291
selecting command processing 8
sequence number 36, 112
session, DFSMSrmm
  starting 1
  stopping 11
setting
  date format 7
  expiration date 59
  home location 70
  owner notification 32
  policy requirements 20
  setting a default vital record specification 42
shelf location
  adding information 23, 166, 182
  changing 77
  defining 23, 166, 182
  deleting information 25, 263, 272
shelf location *(continued)*
  details 111, 285, 302
  line operators 123, 124
  list 115, 310, 325
shortcut keys 421
shutting down the DFSMSrmm subsystem
  operator system error message issued 146
  process for 134
software product
  adding information 33, 179
  changing information 34, 232
  defining 33, 179
  deleting information 34, 270
  details 111, 300
  line operators 123, 124
  list 115, 322
  updating information 34, 232
  volume 33, 34
SORT command
  using 120
sorting
  default sort order 14
  directions 14
  lists 117
  options 117
  order 117
  priorities 14
special dates, support for 59
stacked physical volume, changing 259
START command 131
starting
  DFSMSrmm dialog 1
  DFSMSrmm subsystem 131
starting a search for volume information 334
status for a volume 26
STGADMIN.EDG.IGNORE.TAPE 29
STGADMIN.EDG.LISTCONTROL 113, 287
STGADMIN.EDG.MASTER 29
STGADMIN.EDG.RELEASE 275
sticky label
  default label description 160
  processing 160
stopping
  DFSMSrmm dialog 11
  DFSMSrmm subsystem 134
storage class recorded by DFSMSrmm 35
storage group recorded by DFSMSrmm 35
storage location
  adding bin number 24
  deleting bin number 25
  moving from 79
  moving volumes 55
  retaining volumes 55
  shelf locations 23
submitting
  batch job 12, 165
SUPPORT fast path command 417
support for OPC restart capability to mark DFSMSrmm-managed datasets incorrect before restarting jobs 221
SUPPORT operand 3

system error messages 146
system options 113
system-managed tape library
  adding volumes example 204
  ejecting volumes 76, 342
  freeing up space 71
  moving volumes 55, 71
  moving volumes to 75
  obtaining information about resident volumes 26

T

tape configuration data base
  obtaining information from 26
  updating volume information 76
  validating library name 75
tape label
  expiration date 106
  specifying 194, 335
  type 194, 335
  validation 76
tape librarian
  deleting an owner ID 32
tape mount messages 140
tape processing messages 145
tape validation messages 142
testing
  return codes in a CLIST 355
  return codes in a REXX procedure 355
TRACE operand 3
tracking I/0 errors 103
transferring volume ownership 32, 268
TSO subcommand
  abbreviations 163, 164
  ADDBIN subcommand 166
  ADDDATASET subcommand 169
  ADDOWNER subcommand 176
  ADDPRODUCT subcommand 179
  ADDRACK subcommand 182
  ADDVOLUME subcommand 186
  ADDVRS subcommand 205
  batch job 165
  CHANGEDATASET subcommand 219
  CHANGEOWNER subcommand 229
  CHANGEPRODUCT subcommand 232
  CHANGEVOLUME subcommand 234
  DELETEBIN subcommand 263
  DELETEDATASET subcommand 266
  DELETEOWNER subcommand 268
  DELETEPRODUCT subcommand 270
  DELETERACK subcommand 272
  DELETEVOLUME subcommand 275
  DELETEVRS subcommand 279
  GETVOLUME subcommand 281
  LISTBIN subcommand 285
  LISTCONTROL subcommand 287
  LISTDATASET subcommand 295
  LISTOWNER subcommand 298
  LISTPRODUCT subcommand 300
  LISTRACK subcommand 302
  LISTVOLUME subcommand 304
TSO subcommand *(continued)*
  LISTVRS subcommand 307
  online help 164
  return codes 355
  SEARCHBIN subcommand 310
  SEARCHDATASET subcommand 315
  SEARCHPRODUCT subcommand 322
  SEARCHRACK subcommand 325
  SEARCHVOLUME subcommand 329
  SEARCHVRS subcommand 347
  specifying 163
  using 163
  variables 367
types of retention 55

U

uncataloging a data set 37
undoing movement and action confirmation 237
until expired
  closed data sets retained 57
  data sets retained until expired 56, 57
  data sets retained while open 57
  merging UNTILEXPIRED and WHILECATALOG 61
  relationship with COUNT and WHILECATALOG operands 57
  retention by open status 57
  retention by volume expiration date 57
updating
  data set information automatically 21
  data set information with RMM CHANGEDATASET subcommand 36, 219
  owner information 19, 32, 229
  software product information 34, 232
  vital record specification 52
  volume information 18, 29, 234
USER fast path command 417
user menus 4
USER operand 3
user status volumes, controlling tape initialization 154
user volume 26
  definition 27
  overwriting 26
  searching for information about 319
user, general 1
using the action bar 9
using the EDGUX100 installation exit 29

V

validating
  library name 75
variables, TSO subcommand 367
virtual tape server
  adding an exported logical volume 192
  building an export/import list 343
  changing logical volume location 247
  changing volume type 259
  ejecting a logical volume 249
  identifying logical volumes to DFSMSrmm 343
vital record group 41

vital record management
See vital record specification
vital record specification
ABEND 57
chaining 39
changing 52
cleaning up name vital record specifications 89
creating list 115, 347
data set
adding 51, 205
changing 52
deleting 279
details 111, 307
list 115, 347
retention 58
defining 51, 205
deleting 52, 279
examples
chaining vital record specifications 40
line operators 123, 124
linking 205
managing vital records 55
matching data set names 43, 44
matching job names 43, 44
matching order 43, 44
matching types 295
merging UNTILEXPIRED and WHILECATALOG retention policies 61
name
adding 51, 205
deleting 53, 279
details 111, 307
list 347
OPEN 57
policy requirements 20
requesting information 111, 307
retaining DFSMShsm ABARS backups 217
retaining DFSMShsm backup and migration tapes 217
setting an installation default 42
updating 52
using 55
volume
adding 51, 205
deleting 279
details 111, 307
list 115, 347
retention 58
vital record specification management value
defining management and retention policy 205
displaying 295
matching with data set names 44
specifying 59, 211
VOL1
clearing the label information 258
searching for ISO/ANSI label version information 338
VOL1 label, specifying ISO/ANSI label version 200
volume
actions list 127
adding duplicate 203
volume *(continued)*
adding duplicate volume serial number 28
adding information 27, 28, 186
adding WORM tapes 186
automatic recording of information 25
calculating usage 306
canceling outstanding moves 238
changing 238, 253
changing information 234
changing information for multiple volumes 30, 31
changing volume type 259
changing WORM tape status 238
clearing location information 247
confirming moves 79, 126
confirming return to scratch activities 247
creating list 329
date assigned to a user 190, 192, 248
defining 25, 186
deleting information 31, 275
details 111, 304
ejecting 76
eligible for release 102
erasing 104, 149
grouping 21
high security 104
in transit 79
initial volume serial number 28
initializing 104, 149
limiting the search for volume information 334
line operators 123, 124, 128
list 115
loan location 78
manually releasing 19, 109
marking a stacked volume closed 245
master status 26
movements list 127
moving 55, 70
non-restricted operands or fields 235
non-scratch
adding 27
changing information 18, 29
obtaining information from the TCDB 26
overwriting
master volumes 26
user volumes 26
preventing a stacked volume from moving 255
private 26
reclaiming from pending status or scratch status 236
redefining duplicate volume serial number 29
release actions 19, 102
release date 18, 106
releasing 102, 275
removing 255
replacing 103
requesting movement 74
requirements for pooling 21
resetting the ABEND status flag 253
restricted operands or fields 235
retained by job name 58
retained until expired 56, 57

volume *(continued)*
retained while cataloged 56
retained while open 57
retaining
by generic volume serial number 50
by specific volume serial number 50
retention date 65
returning from a loan location 79
returning to home location 111
returning to owner 104
returning to scratch status 103
scratch volume 26
software product 33, 34
status 26
tracking I/0 errors 103
transferring ownership 32
types of movement 55
updating information 234
user 26
using CHANGEVOLUME subcommand to confirm volume movement 236
with permanent I/0 errors 103
with the same volume serial number 28, 29
volume catalog
obtaining information about volumes 204
VOLUME fast path command 417
VOLUME operand 3
volume serial number
definition 50
generic volume serial number 50
volume usage count 306
volume usage value 306
volume vital record specification
adding 51
definition 39
VRS fast path command 417
VRS operand 3
VRSBYJOBNAME
defining a vital record specification 58
defining job names 45
filtering with job name masks 47

W

waiting requests, number of 133
while cataloged
catalog date format 66
data sets retained while cataloged 56
merging WHILECATALOG and UNTILEXPIRED 61
minimum retention with WHILECATALOG 57, 216
vital record processing for 61
with special dates 217
WORM tape
adding WORM tapes 186
changing WORM tape status 238
identifying WORM tapes ready for destruction 238
wrong data set name, changing 224
wrong label processing 154
wrong volume serial number, changing 253

Readers' Comments — We'd Like to Hear from You

z/OS
DFSMSrmm Guide and Reference

Publication No. SC26-7404-04

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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

Name _____ Address _____

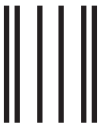
Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC26-7404-04

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

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

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in USA

SC26-7404-04