z/OS
2.5

DFSMSrmm Managing and Using Removable Media

SC23-6873-50

Note

Before using this information and the product it supports, read the information in “Notices” on page 519.

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2023-06-26

© Copyright International Business Machines Corporation 1992, 2022.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ..... **xi**

Tables ..... **xvii**

About This Document ..... **xix**
Required product knowledge xix
z/OS information xix
Notational conventions xix
How to read syntax diagrams xix
How to abbreviate commands and operands xxii
How to use continuation characters xxii
Delimiters xxii
Character sets xxii

How to send your comments to IBM ..... **xxv**
If you have a technical problem xxv

Summary of changes ..... **xxvii**
Summary of changes for z/OS Version 2 Release 5 xxvii
Summary of changes for z/OS Version 2 Release 4 xxvii
Summary of changes for z/OS Version 2 Release 3 xxviii

Chapter 1. Getting started with DFSMSrmm **1**
Using the dialog 1
Starting the dialog 1
User menus 3
Command menu 4
Function menus 4
Getting information about your resources 5
Changing options 6
Displaying options 8
Navigating through DFSMSrmm 9
Using "point-and-shoot" fields 9
Issuing fast path commands 10
Requesting help 11
Exiting the dialog 11
Using lists 12
Using the SELECT command 13
Issuing the SORT command 13
Using SORT options panels 13
Changing the default sort order 14
Printing display lists and panels 15
Requesting batch details 15
Using volumes 16
Requesting a scratch volume 16
Changing volume information 17
Changing the release date for a volume 17
Setting release actions for a volume 18
Changing owner information 18

Releasing volumes manually....18
Requesting notification of release....19
Requesting return after release....19
Retaining data sets and volumes....19
Grouping volumes....20
Understanding Volume Usage, Capacity, and Compression....21
Using tapes created by programs providing their own tape management....22
Changing data set information....23

Chapter 2. Defining your resources using the DFSMSrmm ISPF dialog....25
Defining shelf locations....25
Adding new rack and bin numbers....25
Deleting rack and bin numbers....26
Defining volumes....27
Specifying media type and recording format for volumes....28
Adding a new, non-scratch volume....29
Adding scratch volumes....30
Adding volumes with duplicate volume serial numbers....30
Redefining a volume already defined to DFSMSrmm....30
Changing volume information....31
Changing information for volumes in multivolume sets....31
Rules for changing volume information....32
Deleting volume information....34
Deleting information for volumes in multivolume sets....34
Reclaiming volumes from pending status or scratch status....34
Changing volume location....35
Confirming volume moves and release actions....36
Overriding automatic move processing....36
Canceling outstanding volume moves....37
Changing volume serial numbers....37
Identifying WORM tapes ready for destruction or reuse....37
Changing the status of WORM tapes....38
Defining owners....38
Adding new owners....38
Changing owner information....38
Deleting owner information....39
Defining software products and product versions....39
Adding a new software product....39
Adding software product volumes....40
Changing software product information....40
Deleting software product information....41
Defining data sets....41
Adding new data set information....42
Changing data set information....42
Deleting data set information....42

Chapter 3. Retention methods....45
Retention of multivolume sets....45
The expiration date....45
Expiration based on last reference....47
Summary of the tasks to perform....47
Deciding between the EXPDT and VRSEL retention methods....48
Candidates for exploitation of the EXPDT retention method....49
Changing retention method to EXPDT for existing VRS retained data....50

Chapter 4. Defining vital record specifications....53
Types of vital record specifications....53

Data set vital record specifications....54
Name vital record specifications....54
Volume vital record specifications....54
Chaining vital record specifications....54
Identifying data sets for retention....55
Fully qualified data set names....56
Data set name masks....56
Job names....58
Job name masks....58
Management class and management value names....59
Special ABEND, DELETED, and OPEN retention....59
Name filtering with data set name masks....61
How DFSMSrmm uses jobname to apply policies....62
Name filtering with job name masks....62
How DFSMSrmm applies retention and movement policies....63
Generation data group (GDG) base names....66
Pseudo-GDG data set names....67
Specifying GDG and non-GDG data set names....68
Identifying volumes for retention....68
Retaining volumes by specific volume serial number....69
Retaining volumes by generic volume serial number....69
Adding vital record specifications using the DFSMSrmm ISPF dialog....69
Adding data set vital record specifications....69
Adding volume vital record specifications....69
Adding name vital record specifications....70
Changing vital record specifications....70
Deleting a vital record specification....70

Chapter 5. Using vital record specifications to retain and move volumes....73
Types of retention....73
CYCLES....73
BYDAYSCYCLE....73
DAYS....74
EXTRADAYS....74
LASTREFERENCEDAYS....74
WHILECATALOG....74
UNTILEXPIRED....75
By data set status (open, deleted, or abended)....75
By job name....76
By specific volume serial number....76
By generic volume serial number....76
By specific date....77
Defining retention policies for data sets and volumes....77
Defining policies for management class and management value....78
Combining retention policies in a vital record specification....78
Retaining and moving volumes as sets or individually....79
Defining release policies....79
Defining system-wide vital record specifications....80
DFSMSrmm retention dates....80
Retention date format....81
Special cycles date format....81
Special catalog date format....81
Special CATRETPD date format....81
How DFSMSrmm calculates the retention date....82
Data set retention date....82
Volume retention date....82
Retaining data sets in closed-cycle GDGs....86

Managing closed-cycle GDGs..... 86
Converting from closed-cycle gdgs to using scratch pools.....87
Converting from expiration date or cycle control to using catalog control 87
Defining movement policies for volumes.....88
Moving physical volumes between system-managed libraries..... 88
Managing the contents of system-managed libraries..... 88
Moving to storage locations..... 90
Using manual move control..... 90
Manually requesting volume moves.....91
Requesting a single volume move..... 92
Moving volumes to a system-managed tape library.....92
Ejecting volumes from a system-managed tape library.....93
Moving volumes between pools.....94
Moving volumes to loan locations.....94
Returning volumes from loan locations..... 95
Moving volumes from storage locations..... 95
Confirming volume movements to DFSMSrmm..... 95
Confirming Volume Moves into an Automated Tape Library..... 96
Confirming Volume Moves for Other Locations..... 96

Chapter 6. Using vital record specifications to define retention and movement policies.....99
Before you start defining retention and movement policies..... 99
A summary of the tasks to perform..... 99
Where to find information..... 99
What vital records selection processing you can specify.....100
Chaining retention and movement policies..... 101
Separating the data set name filter from the policy itself.....102
Specifying policies.....103
Defining the minimum required policy.....104
Writing your own vital record specifications.....105
Example 1: Retaining catalogued data sets..... 105
Example 2: Retaining uncatalogued data sets..... 105
Example 3: Retaining cycles of non-GDG data sets.....105
Example 4: Moving data sets to different locations.....106
Example 5: Retaining data sets created within a time period..... 106
Example 6: Holding data sets for extra days.....106
Example 7: Retaining generation data group data sets..... 107
Example 8: Retaining recently used data sets..... 107
Example 9: Using management value to retain data sets.....107
Example 10: Returning volumes to scratch status..... 108
Example 11: Retaining data sets using expiration date.....108
Example 12: Combining vital record specifications.....109
Maintaining your vital record specifications..... 109
EDGRVCLN REXX procedure to clean up vital record specifications.....110

Chapter 7. Requesting and releasing volumes.....113
Requesting scratch volumes manually..... 113
Releasing volumes.....113
Setting release actions for a volume..... 114
Changing a volume's release date..... 118
Confirming manual release actions to DFSMSrmm..... 119
Releasing volumes manually..... 120
Copying and moving tape data.....122
Using the CHANGEDATASET COPYFROM subcommand.....122
Using the EDG_EXIT100 installation exit.....123
Copying the data set attributes..... 123

Retention of the target data set....125
Retention of source data sets managed by VRSEL retention method....125
Retention of source data sets managed by EXPDT retention method....126

Chapter 8. Requesting information about your resources....129
Using DFSMSrmm reporting utilities and sample reports....129
Displaying details about a single resource....129
Managing SEARCH with large results lists....130
Displaying DFSMSrmm system options....131
Using resource lists....132
How to request a list....132
How to sort a list....134
How to Use Line Operators....142
Requesting specific types of lists....143
Scratch volume pull lists....143
Dropship lists....145
Lists of pools defined by your installation....146
Lists of volume movements and actions....146
Creating CLISTs of executable subcommands....146
Printing resource information....148
ISPF data set list utility support....148
ISMF mountable tape volume list support....149
Time zone considerations....149

Chapter 9. DFSMSrmm operator procedures....151
Restarting the DFSMSrmm subsystem....151
Using the START command....151
Using the MODIFY command....152
Stopping the DFSMSrmm subsystem....158
Controlling the Problem Determination Aid (PDA) Facility....160
Responding to DFSMSrmm operator messages....160
Initialization messages....161
Tape mount messages....165
Tape validation messages....167
Tape processing messages....169
System error messages....171
Initializing, erasing, and scanning tape volumes manually....172
Using the LABEL procedure to request EDGINERS processing....172
Replying to LABEL procedure messages....178
Initializing or erasing tape volumes....181
Scanning a tape volume label....184
Processing sticky labels....185
Labels....185
Managing DFSMSrmm installation exits....186

Chapter 10. Using RMM TSO subcommands....187
Issuing the RMM TSO command and subcommands....187
Requesting help for RMM TSO subcommands....188
Using TSO Subcommands from an operator console....188
Submitting a batch job....188
Using TSO subcommands from a TSO CLIST or REXX exec....189
Using RMM TSO subcommands with System REXX....189
ADDBIN: Adding a bin number in a storage location....190
ADDDATASET: Adding data set information....192
ADDOWNER: Adding owner information....202
ADDPRODUCT: Adding software product information....205
ADDRACK: Adding a shelf location....208

ADDVOLUME: Adding volume information....211
ADDVRS: Adding a vital record specification....236
CHANGEDATASET: Changing data set information....251
CHANGEOWNER: Changing owner information....265
CHANGEPRODUCT: Changing software product information....268
CHANGEVOLUME: Changing volume information....269
CHANGEVRS: Changing information about a vital record specification....302
DELETEBIN: Deleting bin number information....305
DELETEDATASET: Deleting data set information....307
DELETEOWNER: Deleting owner information....309
DELETEPRODUCT: Deleting software product information....310
DELETERACK: Deleting shelf location information....311
DELETEVOLUME: Deleting volume information....314
DELETEVRS: Deleting vital record specifications....318
GETVOLUME: Requesting and assigning scratch volumes....320
LISTBIN: Displaying information about a shelf location....325
LISTCONTROL: Displaying parmlib options and control information....327
LISTDATASET: Displaying information about a data set....337
LISTOWNER: Displaying information about an owner....340
LISTPRODUCT: Displaying information about a software product....341
LISTRACK: Displaying information about a shelf location....342
LISTVOLUME: Displaying information about a volume....344
LISTVRS: Displaying information about a vital record specification....349
SEARCHBIN: Creating a list of bin numbers....351
SEARCHDATASET: Creating a list of data sets....357
SEARCHOWNER: Searching owner information....370
SEARCHPRODUCT: Creating a list of software products....373
SEARCHRACK: Creating a list of shelf locations....376
SEARCHVOLUME: Creating a list of volumes....380
SEARCHVRS: Creating a list of vital record specifications....412

Chapter 11. DFSMSrmm return codes and reason codes....423
RMM TSO subcommand return codes....423
RMM TSO subcommand reason codes....424

Chapter 12. DFSMSrmm REXX variables....431
TSO subcommand variables by subcommand....432
TSO subcommand variables by name....442

Appendix A. DFSMSrmm TSO subcommands....467
ADDBIN: Adding a bin number in a storage location....467
ADDDATASET: Adding data set information....467
ADDOWNER: Adding owner information....469
ADDPRODUCT: Adding software product information....469
ADDRACK: Adding a shelf location....469
ADDVOLUME: Adding volume information....470
ADDVOLUME: Volume operands....470
ADDVOLUME: Non-SCRATCH volume optional operands....475
ADDVRS: Adding a vital record specification....477
ADDVRS: Adding a data set vital record specification....477
ADDVRS: Adding a location name vital record specification....478
ADDVRS: Adding a retention name vital record specification....478
ADDVRS: Adding a volume vital record specification....480
CHANGEDATASET: Changing data set information....480
CHANGEOWNER: Changing owner information....483
CHANGEPRODUCT: Changing software product information....484
CHANGEVOLUME: Changing volume information....484

CHANGEVOLUME: Confirmrelease and global movement operands 484
CHANGEVOLUME: Specific volume optional operands 485
CHANGEVRS: Changing a vital record specification 492
DELETEBIN: Deleting bin number information 492
DELETEDATASET: Deleting data set information 492
DELETEOWNER: Deleting owner information 492
DELETEPRODUCT: Deleting software product information 493
DELETERACK: Deleting shelf location information 493
DELETEVOLUME: Deleting volume information 493
DELETEVRS: Deleting vital record specification information 494
GETVOLUME: Requesting and assigning scratch volumes 495
LISTBIN: Displaying information about a shelf location 496
LISTCONTROL: Displaying parmlib options and control information 496
LISTDATASET: Displaying information about a data set 497
LISTOWNER: Displaying information about an owner 497
LISTPRODUCT: Displaying information about a software product 497
LISTRACK: Displaying information about a shelf location 497
LISTVOLUME: Displaying information about a volume 497
LISTVRS: Displaying information about a vital record specification 498
SEARCHBIN: Creating a list of bin numbers 498
SEARCHDATASET: Creating a list of data sets 498
SEARCHOWNER: Creating a list of owner information 501
SEARCHPRODUCT: Creating a list of software products 502
SEARCHRACK: Creating a list of shelf locations 502
SEARCHVOLUME: Creating a list of volumes 502
SEARCHVRS: Creating a list of vital record specifications 510

Appendix B. DFSMSrmm ISPF dialog fast path commands 513

Appendix C. Accessibility 517

Notices 519
Terms and conditions for product documentation 520
IBM Online Privacy Statement 521
Policy for unsupported hardware 521
Minimum supported hardware 521
Programming interface information 522
Trademarks 522
Glossary 523

Index 543

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..... 6

7. DFSMSrmm Dialog Options Menu..... 9

8. DFSMSrmm Volume Display panel..... 11

9. DFSMSrmm Exit Menu.....12

10. DFSMSrmm Dialog Sort Options Menu.....14

11. DFSMSrmm Data Set List Sort Options panel..... 14

12. DFSMSrmm Volume List Sort Options panel.....15

13. DFSMSrmm processing multiple volumes..... 32

14. Chaining vital record specifications.....55

15. Specifying a name vital record specification for a chain.....55

16. Specifying a vital record specification chain..... 55

17. Valid job name masks..... 59

18. Defining a retention policy for a GDG base name..... 66

19. Duplicate GDG example.....66

20. Specifying data set name masks for pseudo-GDGs.....67

21. Specifying data set name masks to manage data sets like a GDG.....68

22. Specifying a global data set name mask to manage GDGs.....68

23. Specifying NOGDG and CYCLES in a vital record specification.....68

24. Specifying NOGDG and DAYS in a vital record specification....68
25. Retaining a data set using management class....78
26. Retaining a data set using management value....78
27. Specifying combinations of retention policies on a single vital record specification....79
28. Using UNTILEXPIRED and WHILECATALOG on a single vital record specification....79
29. Managing closed-cycle GDGs....87
30. Managing volumes using scratch pools....87
31. Managing volumes using catalog control....87
32. Moving volumes out of a system-managed library....89
33. Moving a volume in and out of the system-managed library....89
34. Moving data out of the system-managed library at fixed intervals....90
35. Keeping volumes on-site....91
36. Sending a volume to another system....91
37. Returning a volume from another system....91
38. Ejecting a volume from a system-managed library....93
39. Moving a volume to a loan location....95
40. Supplying loan location information for volumes....95
41. Returning a volume from a loan location....95
42. DFSMSrmm Volume Action Status panel....96
43. DFSMSrmm Volume Action Summary List....97
44. Confirming volume moves for volumes ready to scratch....98
45. Identifying locations and destinations for volumes....98
46. Using special ABEND, DELETED, or OPEN job names....100
47. Specifying a vital record specification with default operands....103
48. Specifying a vital record specification using the dialog part 1....104

49. Specifying a vital record specification using the dialog part 2 104

50. Sample JCL to run the EDGRVCLN procedure 112

51. Confirming replace actions for volumes 120

52. DFSMSrmm Volume Display panel 130

53. DFSMSrmm System Options Display 132

54. DFSMSrmm Dialog Sort Options Menu 135

55. DFSMSrmm Data Set List Sort Options panel 135

56. SORT command syntax diagram for rack or bin number lists 140

57. SORT command syntax diagram for volume lists 140

58. SORT command syntax diagram for lists of volume movements and actions 141

59. SORT command syntax diagram for software product 141

60. SORT command syntax diagram for data set lists 141

61. SORT command syntax diagram for vital record specification lists 142

62. SORT command syntax diagram for report generator lists 142

63. Producing a list of scratch volumes 143

64. DFSMSrmm Volume Search panel 144

65. DFSMSrmm Volume Action Status panel 145

66. DFSMSrmm Time Zone Help panel 150

67. START command syntax diagram 151

68. MODIFY command syntax diagram 153

69. Output from QUERY ACTIVE command 157

70. LABEL command syntax diagram 173

71. ERASE or INIT request syntax diagram 182

72. SCAN request syntax 184

73. Default label format for a tape cartridge 185

74. Default label format for a round tape....186

75. Sample LISTBIN output....326

76. Sample LISTCONTROL STATUS output....331

77. Sample LISTCONTROL CNTL OPTION output (Part 1 of 2)....331

78. Sample LISTCONTROL CNTL OPTION output (Part 2 of 2)....332

79. Sample LISTCONTROL output - options and rules, excluding control record information and start up options....333

80. Sample LISTCONTROL PRTITION output....333

81. Sample LISTCONTROL OPENRULE output....334

82. Sample LISTCONTROL SECLEVEL output....334

83. Sample LISTCONTROL LOCDEF output - No LOCDEFs defined....334

84. Sample LISTCONTROL MEDINF output....335

85. Sample LISTCONTROL MEDINF output - No media information defined....335

86. Sample LISTCONTROL LOCDEF output - LOCDEFs defined....336

87. Sample LISTCONTROL DEFTABLE output....336

88. Sample LISTDATASET output....339

89. Sample LISTOWNER output....340

90. Sample LISTPRODUCT output....342

91. Sample LISTRACK output....344

92. Sample 1: LISTVOLUME output....346

93. Sample 2: LISTVOLUME output....347

94. Sample 3: LISTVOLUME output....348

95. Listing vital record specification information examples....350

96. Sample LISTVRS output....350

97. Sample SEARCHBIN output for empty bins....355

98. Sample SEARCHBIN output....355

99. Sample SEARCHBIN output for bins numbers using the CONTINUE operand....356

100. Sample SEARCHBIN output for bins numbers using the CONTINUE operand....356

101. Sample SEARCHBIN output for bins numbers using the CONTINUE operand....356

102. Sample SEARCHDATASET output....369

103. Sample SEARCHDATASET output....369

104. Sample SEARCHDATASET output....369

105. Sample SEARCHDATASET output....370

106. Sample CLIST data set....370

107. Sample SEARCHOWNER output listing....373

108. Sample SEARCHPRODUCT output....376

109. Sample SEARCHRACK output....380

110. Sample SEARCHVOLUME output listing duplicate volumes....408

111. Sample SEARCHVOLUME output listing owned volumes....408

112. Sample SEARCHVOLUME output listing volumes to eject....409

113. SEARCHVOLUME output using CLIST operand....409

114. Sample SEARCHVOLUME output using RETDATE operand....409

115. Sample SEARCHVOLUME output building a list of logical volumes....409

116. Sample SEARCHVOLUME CLIST output for a list of logical volumes....409

117. Sample SEARCHVOLUME CLIST output identifying logical volumes to DFSMSrmm....410

118. Moving volumes from one library to another library....410

119. Confirming volume moves from one library to another library....410

120. Listing volumes within a range of volume serial numbers....411

121. Using a generic volume serial number to obtain a list of volumes within a range of volume serial numbers....411

122. Listing volumes for an owner within a range of volume serial numbers....411

123. Listing volumes for the export list....411

124. CLIST information for the export list....412
125. Sample SEARCHVRS output....420
126. Sample SEARCHVRS output....421
127. Sample SEARCHVRS output....421
128. Testing RMM TSO LISTVOLUME subcommand example 1....423
129. Testing RMM TSO LISTVOLUME subcommand example 2....423

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 28

6. How DFSMSrmm assigns media type and recording format 28

7. How DFSMSrmm uses jobname to apply policies 62

8. How DFSMSrmm matches job names to job name masks 62

9. How DFSMSrmm matches data set names to data set masks 64

10. Subchain used for different GDG DUPLICATE options 67

11. Retention date format displayed 81

12. DFSMSrmm retention date calculation by COUNT from 1 through 99998 83

13. DFSMSrmm retention date calculation by COUNT(99999) 84

14. Release actions for a volume 114

15. Confirm list line operators 120

16. DFSMSrmm release volumes line operators 121

17. Data set attributes that are not copied by COPYFROM 124

18. Information to specify on display panels 129

19. How to display search panels 133

20. Table field names on DFSMSrmm lists 136

21. Operator messages 160

22. Operator responses to message EDG1107D 164

23. Operator responses to message EDG6663D 166

24. Operator responses to message EDG6663D..... 180

25. RMM TSO subcommands.....187

26. New owner information..... 205

27. Assigning rack numbers to volumes that DFSMSrmm manages.....208

28. How DFSMSrmm assigns rack numbers to a volume..... 212

29. LISTCONTROL Subcommand output.....335

30. Information returned by SEARCHBIN..... 351

31. Information returned by SEARCHDATASET..... 357

32. Information returned by the RMM SEARCHOWNER subcommand.....371

33. Information returned by SEARCHPRODUCT..... 374

34. Information returned by SEARCHRACK..... 377

35. Information returned by the RMM SEARCHVOLUME subcommand.....381

36. Creating lists of vital record specifications..... 412

37. Information returned by SEARCHVRS.....413

38. DFSMSrmm reason codes.....424

39. TSO subcommand variables by RMM subcommand.....432

40. TSO subcommand variables by name..... 442

41. DFSMSrmm ISPF dialog fast path commands..... 513

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, for users who have a physical disability, see Appendix C, “Accessibility,” on page 517.

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

z/OS information

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

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

To find the complete z/OS library, go to IBM Documentation (www.ibm.com/docs/en/zos).

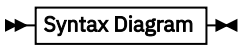
Notational conventions

This section explains the notational conventions used in this document.

How to read syntax diagrams

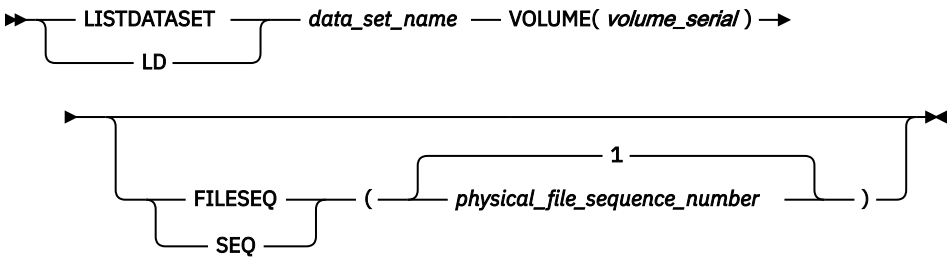
Throughout this library, diagrams are used to illustrate the programming syntax. Keyword parameters are parameters that follow the positional parameters. Unless otherwise stated, keyword parameters can be coded in any order. The following list tells you how to interpret the syntax diagrams:

- 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.

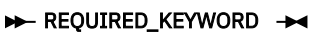


- If a diagram is longer than one line, each line to be continued ends with a single arrowhead and the next line begins with a single arrowhead.

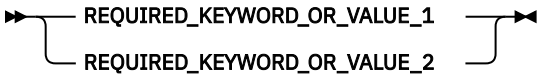
© Copyright IBM Corp. 1992, 2022



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

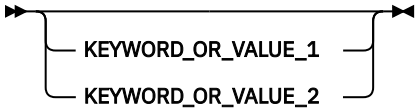


  If several mutually exclusive required keywords or values exist, they are stacked vertically in alphanumeric order.



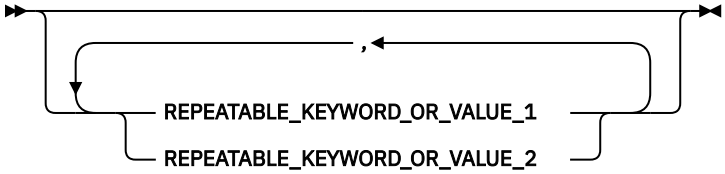
- Optional keywords and values appear below the main path line. You can choose not to code optional keywords and values.

  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 separated from the next by a comma.

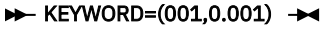
- An arrow returning to the left above a group of keywords or values means more than one can be selected, or a single one can be repeated.



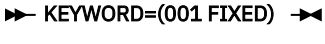
- A word in all uppercase is a keyword or value you must spell exactly as shown. In this example, you must code ***KEYWORD***.

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

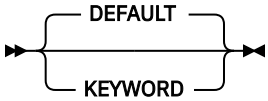
- If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code ***KEYWORD=(001,0.001)***.

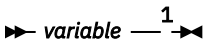
- If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code ***KEYWORD=(001 FIXED)***.

- 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:

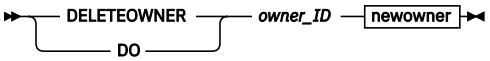
  1 An example of a syntax note.

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

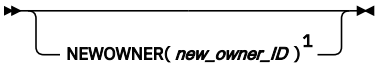
- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

  Syntax Fragment

The following is an example of a syntax diagram.



newowner



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 on page xxii. Table 2 on page xxii lists the special characters that have syntactical functions in job control statements.

Table 1. Character sets

Character Set	Contents	Notes®
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

Table 2. Special characters used in syntax (continued)	
Character	**Syntactical function**
=	To separate a keyword from its value, for example, BURST=YES
(ḃ)	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
*	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

How to send your comments to IBM

We invite you to submit comments about the z/OS product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page xxv.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM® Documentation function
If your comment or question is about the IBM Documentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS DFSMSrmm Managing and Using Removable Media, SC23-6873-50
- The section title of the specific information to which your comment relates
- The text of your comment.

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

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

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

© Copyright IBM Corp. 1992, 2022

Summary of changes

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

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www-01.ibm.com/servers/resourcelink/svc00100.nsf/pages/ibm-zos-doc-update-policy?OpenDocument).

Summary of changes for z/OS Version 2 Release 5

The following changes are made for z/OS Version 2 Release 5 (V2R5). The most recent updates are listed at the top of each section.

Changed

The following content is changed.

May 2023 refresh

- In support of APAR OA64283 "TSO subcommand variables by name" on page 442, "TSO subcommand variables by subcommand" on page 432 and "LISTCONTROL: Displaying parmlib options and control information" on page 327 are updated.(Which also applies to z/OS V2R4.)
- RELEASE(EXPIRYDATEIGNORE,SCRATCHIMMEDIATE) is updated in "ADDVRS: Adding a vital record specification" on page 236 and notes are added to Table 12 on page 83 and Table 13 on page 84.

November 2022 refresh

- In support of APAR OA62962, "The expiration date " on page 45 is updated. (APAR OA62962, which also applies to z/OS V2R4 and V2R3.)

Prior to November 2022 refresh

- DFSMSrmm no longer supports CIM and any information is removed.

Summary of changes for z/OS Version 2 Release 4

The following changes are made for z/OS Version 2 Release 4 (V2R4). The most recent updates are listed at the top of each section.

New

Prior to June 2020 refresh

The following content is new.

- Added new tape processing message EDG4064D. For more information, see "Responding to DFSMSrmm operator messages" on page 160.
- Added new tape validation messages EDG4065I, EDG4066I, and EDG4067I. For more information, see "Tape validation messages" on page 167 and "Responding to DFSMSrmm operator messages" on page 160.
- Added new tape processing message EDG4064D. For more information, see "Tape processing messages" on page 169.
- Added new FORCEEXPIRE and REMOVEEXPIRE parameters. For more information, see "CHANGEDATASET: Changing data set information" on page 251.

© Copyright IBM Corp. 1992, 2022

- Added new EDM and NOEDM parameters. For more information, see “SEARCHVOLUME: Creating a list of volumes” on page 380.
- Added new reason code for EDG3377. For more information, see “RMM TSO subcommand reason codes” on page 424.
- Added new EDG@DSXP variable to the LISTCONTROL OPTION subcommand, EDG@FRXP to the LISTDATASET subcommand, EDG@CDSF to the LISTCONTROL CNTL subcommand, and EDG@DEFM, EDG@DEFN, EDG@DEFP, and EDG@DEFZ to the LISTCONTROL DEFTABLE subcommand. For more information, see “TSO subcommand variables by subcommand” on page 432.
- Added new variables EDG@CDSF, EDG@CTRT, EDG@DEFM, EDG@DEFN, EDG@DEFP, EDG@DEFS, EDG@DEFZ, EDG@DSXP, and EDG@FRXP. For more information, see “TSO subcommand variables by name” on page 442.

Changed

Prior to June 2020 refresh

The following content is changed.

- Changed by adding FORCEEXPIRE and NOFORCEEXPIRE to the SEARCHDATASET subcommand syntax. For more information, see “SEARCHDATASET: Creating a list of data sets” on page 357.
- Changed by adding EDM and NOEDM to the SEARCHVOLUME subcommand syntax. For more information, see “SEARCHVOLUME: Creating a list of volumes” on page 380.
- Changed by adding FORCEEXPIRE and REMOVEEXPIRE to the CHANGEDATASET subcommand syntax. For more information, see “CHANGEDATASET: Changing data set information” on page 251.
- Changed by updating the Volume Search panel. For more information, see “Using the Volume Search panel” on page 143.
- Changed by adding information regarding the case where for a specific mount request of a volume not defined to DFSMSrmm. For more information, see “Tape validation messages” on page 167.
- Changed by adding 'ForceExpire' to the Format column. For more information, see “TSO subcommand variables by name” on page 442.
- Changed by adding the FORCEEXPIRE flag to the LISTDATASET command. For more information, see “LISTDATASET: Displaying information about a data set” on page 337.
- Changed by adding the CDSFULL threshold to the LISTCONTROL CNTL output, and parameters to LISTCONTROL DEFTABLE output. For more information, see “LISTCONTROL: Displaying parmlib options and control information” on page 327.

Deleted

No content was removed from this information.

Summary of changes for z/OS Version 2 Release 3

This edition contains updates for Version 2 Release 3 (V2R3).

New information

This edition includes the following new information:

- New EXPTM and WHILECATALOG operands have been added to “ADDDATASET: Adding data set information” on page 192.
- New EXPTM operand has been added to “ADDVOLUME: Adding volume information” on page 211 and “GETVOLUME: Requesting and assigning scratch volumes” on page 320.
- “TSO subcommand variables by name” on page 442 contains new Catalog Retained REXX information.
- New DEFTABLE operand has been added to “LISTCONTROL: Displaying parmlib options and control information” on page 327.

- “ADDDATASET: Adding data set information” on page 192
- “ADDVOLUME: Adding volume information” on page 211
- “CHANGEDATASET: Changing data set information” on page 251
- “CHANGEVOLUME: Changing volume information” on page 269
- “DELETEVOLUME: Deleting volume information” on page 314
- “GETVOLUME: Requesting and assigning scratch volumes” on page 320
- “LISTCONTROL: Displaying parmlib options and control information” on page 327
- “SEARCHDATASET: Creating a list of data sets” on page 357
- “SEARCHVOLUME: Creating a list of volumes” on page 380
- “SEARCHVRS: Creating a list of vital record specifications” on page 412
- “Using tapes created by programs providing their own tape management” on page 22 contains information regarding EDM-managed tapes.
- “Changing volume information based on volume status” on page 33 contains new EDM/NOEDM operand.
- “Reclaiming volumes from pending status or scratch status” on page 34 contains new EDM/NOEDM operand.
- “Releasing volumes” on page 113 contains new EDM volume attribute information.
- “CHANGEVOLUME: Changing volume information” on page 269 contains description of new EDM/NOEDM parameters.
- “TSO subcommand variables by subcommand” on page 432 LISTVOLUME VOL subcommand contains new variable EDM.
- “TSO subcommand variables by name” on page 442 added new EDG@EDM variable.
- “CHANGEVOLUME: Specific volume optional operands” on page 485 updated with new EDM/NOEDM operands.
- “Glossary” on page 523 added new term EDM-managed.

Changed information

This edition includes the following topics that contain changed information:

- “TSO subcommand variables by subcommand” on page 432 added new REXX variables.
- “TSO subcommand variables by name” on page 442 added new REXX variables and new values for EDG@RMSB variable.
- “ADDDATASET: Adding data set information” on page 467
- “ADDVOLUME: Adding volume information” on page 470
- “CHANGEDATASET: Changing data set information” on page 480
- “CHANGEVOLUME: Changing volume information” on page 484
- “GETVOLUME: Requesting and assigning scratch volumes” on page 495
- “SEARCHDATASET: Creating a list of data sets” on page 498
- “TSO subcommand variables by name” on page 442 was updated with new value PERMANENT.
- “Displaying DFSMSrmm system options” on page 131
- “DELETEVOLUME: Deleting volume information” on page 314 changed to include new EDM attribute.
- “RMM TSO subcommand reason codes” on page 424 contains new and changed reason codes.
- “Understanding Volume Usage, Capacity, and Compression” on page 21 contains a note regarding the Percent full report.

Chapter 1. Getting started with DFSMSrmm

This topic 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 topic explains how to use the DFSMSrmm dialog from these 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 on page 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 V2R1
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.
```

Figure 1. DFSMSrmm Primary Option Menu

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 on page 2. See the*z/OS DFSMS Using the Interactive Storage Management Facility* for additional information.

© Copyright IBM Corp. 1992, 2022

```
 Panel  Help
------------------------------------------------------------------------------
EDGP@USR                      DFSMSrmm User Menu - z/OS V2R1
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.
```

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

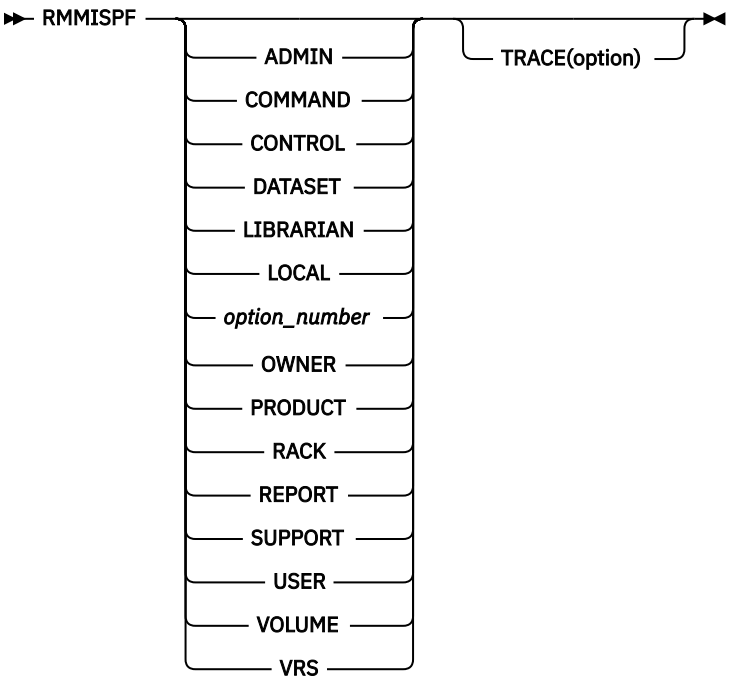


Figure 3. RMMISPF EXEC syntax diagram

Each of the operands, except for TRACE and option_number, represents a menu from which you can request functions. For example, if you enter this 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 these 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 these 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 on page 4 shows the Command Menu DFSMSrmm displays.

```
 Panel  Help
------------------------------------------------------------------------------
EDGP@CMD                    DFSMSrmm Command Menu - z/OS V2R1
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    - Report generator


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

Figure 4. DFSMSrmm Command Menu

Function menus

DFSMSrmm records information about these 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 on page 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.
```

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 on page 5 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 detailed information about the volume.
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.

Table 3. Requesting lists and display panels from the User Menu (continued)

Type of Resource	Requesting Lists	Requesting a Display Panel
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 6.

1. To change processing options, select Option 1 (USER) on the Dialog Options Menu and press ENTER. Figure 6 on page 6 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 )
Time zone . . . . . . MESZ +01:00:00   ( zone offsetHH:MM:SS)

Confirm deletes . . . YES              ( Yes or No )

Processing option . . F                 F - Foreground, B - Background
DSNAME case option  . U                 M - Mixed, U - Upper
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 these dialog options:

- What time zone offset should be used when date and time values are specified
- What date format should be used with lists and panels
- Whether 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
- What case option (upper or mixed) should be used with Data Set Name.
- 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 on page 7 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	12/15/2011
European	dd/mm/yyyy	15/12/2011
Iso	yyyy/mm/dd	2011/12/15
Julian	yyyy/ddd	2011/349

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 these 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.

Setting the time zone

Choose what time zone offset should be used when date and time values are specified. The format is {+|-}HH[:MM[:SS]] where:

- +|- is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- HH is hours
- MM is minutes
- SS is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

Selecting command processing

You can select how DFSMSrmm is to process your requests for functions. Choose one of these 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 41 for more information on the individual confirm delete panels.

Specifying foreground or background processing

You can specify whether DFSMSrmm processes your requests interactively (in the foreground) or saves them in a data set to be processed when you exit the dialog (in the background). Enter F in Processing option for foreground processing. Enter B in Processing option for background processing.

Setting the Data Set Name case option

By default the Data Set Name is folded to uppercase. You can change this default processing to enable support for lowercase or mixed case by changing DSNAME case option from U to M.

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 these 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 on page 9.

```
 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.
```

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 142 for information on requesting functions using list line operators from lists, and Chapter 10, “Using RMM TSO subcommands,” on page 187 for information on requesting functions using the TSO command and subcommands.

Using "point-and-shoot" fields

The DFSMSrmm dialog provides selected "point-and-shoot" fields on the Volume and Data Set Detail panels. Examples of these point-and-shoot fields include:

- On Dataset Detail Panel:
  - Primary and secondary VRS fields
  - Volume volser
- On Volume Details Panel:
  - Previous volume
  - Next volume
  - Volume sequence
  - Number of data sets

Point-and-shoot text fields are cursor-sensitive; if you move the cursor over a point-and-shoot field and click on it (or press Enter), the action associated with that field is performed. The point-and-shoot fields

display as push buttons in the ISPF GUI. See *z/OS ISPF User's Guide Vol I* for more information on using point-and-shoot fields.

Note:

1. If you have entered a command on the command line, it is processed before any point-and-shoot command unless you are running in GUI mode.
2. You can use the tab key to position the cursor to point-and-shoot fields by selecting the "Tab to point-and-shoot fields" option on the ISPF Settings panel.
3. Point-and-shoot fields span the output field only; no prompt text included.
4. If a point-and-shoot action is not appropriate due to a null or zero value, no point-and-shoot action is taken. A message "Action not supported" is shown instead.
5. When the point-and-shoot action results in a search, the message "Search in progress" is displayed.
6. To more easily see the fields that are enabled for point-and-shoot, you can customize the color, intensity, and highlighting of the point-and-shoot fields. Issue the ISPF system command PSCOLOR from any ISPF command line and adjust the Point-and-Shoot Panel Element as desired. Refer to *z/OS ISPF User's Guide Vol II* for additional information.

Issuing fast path commands

Related Reading: Use the fast path commands listed in Appendix B, “DFSMSrmm ISPF dialog fast path commands,” on page 513 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 on page 11), 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 these 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 these 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 this format:

```
<tso prefix>.RMMnn.LIST
```

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

Using lists

This topic describes these tasks associated with lists:

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

Using the SELECT command

The SELECT command is supported on all results lists. You can use this primary command to apply the same line command to all selected entries in a search results list table. For example, all volumes in the volume result list are ejected when you issue:

```
Command ===>SELECT * E
```

in a volume search results list.

You can also specify SEL or S, instead of SELECT. When you do not specify a line command, S is the default.

You can specify a generic value for the first parameter, or a specific value. For example, issue:

```
SELECT AB0* V
```

to view volume data for all volumes starting with AB0, or issue:

```
SELECT MW0001 R
```

to release the volume MW0001, if found in the list.

The first parameter is applied on the first key of the table. If you want another column of the table to be taken as the select criteria, issue a SORT command before the select command. For example, first SORT by owner, then issue:

```
SELECT GEORGE E
```

to eject all volumes of owner GEORGE.

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 134 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 these 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 on page 14 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.
```

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 14shows.

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

     Enter the relative sort priority ( 1 to 7 ) and direction ( A or D ):

   Table field name             Priority     Direction
   ----------------             --------     ---------
   Data set name                   1            A
   Volume serial                   2            A
   Owner
   File sequence
   Create date
   Expiration date
   VRS retained




      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 134.

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 these:

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
   Set Retained                          Label
   Status                                Recorded format
   Location                              Compaction
   Destination                           Attributes
   Transit
   Data Sets



      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 Vol I* for more information.

Requesting batch details

You can request lists of resources and details about those resources in batch, by performing these 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 these 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 topic describes these tasks associated with volumes.

For information about	See
Requesting a scratch volume	“Requesting a scratch volume” on page 16
Changing volume information	“Changing volume information” on page 17
Changing the release date for a volume	“Changing the release date for a volume” on page 17
Setting release actions for a volume	“Setting release actions for a volume” on page 18
Changing data set information	“Changing data set information” on page 23
Changing owner information	“Changing owner information” on page 18
Releasing volumes manually	“Releasing volumes manually” on page 18
Requesting notification of release	“Requesting notification of release” on page 19
Requesting return after release	“Requesting return after release” on page 19
Retaining data sets and volumes	“Retaining data sets and volumes” on page 19
Grouping volumes	“Grouping volumes” on page 20
Understanding volume usage, capacity, and compression	“Understanding Volume Usage, Capacity, and Compression” 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 269 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 this 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 if you want DFSMSrmm to return the specified number of volumes. Otherwise, all the volumes that have met the search criteria are returned. 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.

If you want the volume to be considered for release immediately, you can release it manually. See "Releasing volumes manually" on page 18 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 these 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 these 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, or a valid e-mail address.

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 18 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 18 for more information.

If you specified an e-mail address, you must have an SMTP server configured and started. The default SMTP server is SMTP on the current JES node. To configure any other SMTP server for use, see the ADDOWNER command for the restricted owner called 'SMTP'.

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 18 for more information.

Retaining data sets and volumes

Your storage administrator can implement a retention method to retain data sets or volumes you own based on your needs. DFSMSrmm offers two retention methods, EXPDT and VRSEL. See Chapter 3, “Retention methods,” on page 45 for more information about choosing a retention method.

Using the EXPDT retention method

If your volume is retained by the EXPDT retention method, only the desired expiration date must be provided. Only inventory management expiration processing determines whether the volume is retained or expired based on the expiration date. If movement of the volume is desired, it must be a manual move.

When you use the EXPDT retention method:

- The volume attribute RETAINBY determines whether volumes are retained individually or as a set, and whether a volume or volume set should be retained for as long the first data set on it is retained.
- The GDG and NOGDG options can be used to specify default attributes for new GDG (generation data group) and non-GDG data sets. These attributes are:
  - RETPD, which specifies the retention period (in days) for the data set.
  - WHILECATALOG, which specifies whether the data set is to be retained as long as it is cataloged. WHILECATALOG also specifies how the data set's expiration date should be applied in combination with its catalog status. For example, you could specify that a data set should not be expired as long as it is cataloged, nor should it be expired prior to its expiration date.

  The RETPD attribute can also be specified independently of the GDG and NOGDG options, as a global default.
- The optional data set attribute CATLGDAYS can be used to define how many days a data set specified with WHILECATALOG(ON) or WHILECATALOG(UNTILEXPIRED) should be retained after it is uncataloged.
- The optional data set attribute LASTREF can be used to define how many days a data set should be retained after it was last read or written. The expiration date and time will be updated dynamically after each reference to the data set.

See Chapter 3, “Retention methods,” on page 45 for more information on using the EXPDT retention method.

Using the VRSEL retention method

For data sets or volumes that are to be retained by the VRSEL retention method, the storage administrator must define retention policies, known as vital record specifications, for those data sets or volumes. 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 this 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 your data set is on a volume retained by the VRSEL retention method, but you want it to be excluded from the VRSEL processing, provide this information to your storage administrator.

If you want to retain a volume you own, provide this 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 this 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?

Understanding Volume Usage, Capacity, and Compression

DFSMSrmm tracks information about what you write onto tape volumes, including record length, block size and number of blocks. DFSMSrmm also tracks information about the data written to the tape volume, such as the compressed size and physical size of the file and can calculate the compression ratio. In addition, information is recorded about the media capacity, percentage used, physical space used, and the overall compression ratio.

Basic information about the media can be defined by command, but, when a volume is used for output, DFSMSrmm records the media type, recording format, capacity and percentage used. Normally this information is provided by the tape drive when the drive is an IBM 3590, or IBM TS11x0 or later drive. For older tape devices such as 3490, and also for virtual tape emulating 3490, this information is not available and is derived from media information hard-coded in DFSMSrmm. For non-IBM media, and to override these details for IBM media, you can define media information to DFSMSrmm using the MEDINF command in the DFSMSrmm parmlib.

The following information is recorded or calculated:

Volume capacity
The Capacity (MB) field displays the physical capacity of the tape volume as reported by the hardware. The value is displayed in megabytes for those devices which provide capacity information, such as the IBM 3590 Magstar® and later devices. When not reported by the hardware, the capacity is derived from the media type and recording format. These values are hard-coded in DFSMSrmm or can be specified by MEDINF in parmlib.

Percent full/Percent of volume
The Percent full field displays how much of the volume has been taken up by the data written to it. The value is calculated using the reported position on the volume where the end of the last file was written. When not reported by the hardware DFSMSrmm uses the volume capacity and the physical bytes written to the volume after compression to calculate the percent full. When data is written to a volume the bytes written are reported by the hardware in 4KB increments and so, when volume position is not reported, the displayed values are approximations of actual values. The Percent of volume field displays how much of the physical media space is occupied by the data set or file.

Note: If the amount of blocks written onto the volume reaches maximum allowed by the hardware, the volume is treated by the hardware as 100% Percent full. In this case, the DFSMSrmm Percent Full value might differ significantly from the value reported by the hardware.

Compression
The compression field displays the compression ratio for the data your application wrote to the volume. It is calculated using *ApplicationBytesWritten / DeviceBytesWritten* and displayed to an accuracy of 2 decimal places. For example, a compression value of '3.33' is a ratio of 3.33:1. Consider that *ApplicationBytesWritten* is the data set size field, and for RECFM VB is an approximation. Also *DeviceBytesWritten*, displayed as the data set physical size, may be rounded to 1KB or 4KB increments depending on the hardware involved. As a result, the reported compression value can fluctuate significantly with smaller data set sizes. If no compression ratio can be computed, a value of 0.00. is reported.

Data set size
This is an estimation of how much data was written by the application prior to any compression/compaction. The Block Count field specifies the number of blocks written to the tape volume. The block count value displayed corresponds to that recorded in the data set's End-of-File label. The Block Size and the Block Count are used to estimate the Size of the data set.

Note: Starting from z/OS V2R2, the data set size for a data set with RECFM=U is calculated using statistics reported by the hardware. If the hardware doesn't provide required data, then the data set size is calculated using the same algorithm as in lower z/OS releases. Thus, the value calculated under lower releases might differ noticeably from the value calculated when running under z/OS V2R2 and above.

Volume usage
The Volume Usage field displays how much of the volume has been taken up by the data sets that reside on it. The volume usage is the sum of the size of all data sets on the volume.

Physical size
The data set physical size displays how much data is reported to have been written by the tape drive to the volume and is reported after any compression. The values reported depend on the tape hardware used. IBM tape drives defined as 3590-1 report using 1KB increments. Older IBM tape drives such as 3490 and virtual tape emulating 3490 report in 4KB increments.

Physical used
The volume physical used field displays the sum of the physical size of all files on the volume. Many small files on a 3490 can potentially distort the reported values. For IBM tape drives that report using 1KB increments this value accurately reflects the physical tape usage and is not a sum of all the tape files physical size.

Accurate recording of information depends on all files on a volume being created and updated by a z/OS release that supports recording of physical size:

- When data was written prior to the support for tracking physical size/usage DFSMSrmm displays no value (0KB).
- When data was written on a supporting level, a valid physical size is displayed.
- In case of a volume containing a mix of data sets created before the new function and some created with the new function, each individual data set record displays values as listed above, however the displayed volume physical used and percentage full values will be incorrect because they can only reflect the sum of the data set values that are actually available.

LISTVOLUME and LISTDATASET display the values recorded by DFSMSrmm. Refer to “Extract file processing” on page 22for how to get better information.

Extract file processing

When you create an extract file and request extended records, the report extract processing attempts to determine correct values and, if a correct value cannot be calculated or derived, the field is set to **********. The data set fields calculated based on an existing "percent of volume" value are:

- physical size
- compression.

The volume fields which are calculated based on existing "percent full" value are:

- physical used
- compression.

When no percent full/percent of volume is available rmm extract processing cannot calculate missing values for physical size/used or compression and values are set to zero. The affected volume fields when a correct value cannot be derived, because some files were recorded and some were not and percent of volume values are not available, are:

- percent full. Set to ***
- compression. Set to ******
- physical used. Set to **********.

Using tapes created by programs providing their own tape management

DFSMSrmm provides an additional protection for tapes created by programs that provide their own tape management. Such tapes are referred as EDM-managed, that means controlled by External Data Manager. This additional protection affects the DFSMShsm, DFSMSdfp OAM and IBM Spectrum Protect components by preventing tapes from being accidentally released by users they do not belong to. Note that the defaults table can be used to change which volumes are chosen to be EDM-managed.

This includes:

- Setting of the special EDM volume attribute at the OPEN time when a volume is written by the programs ANZSRVR, ANRSERV, ARCCTL, ARCWCTL, CBROAM, DSMSERV and the expiration date specified in the DD-statement is 1999/365, 1999/366 or it is not specified at all.
- The ability to set and reset the EDM attribute with the RMM CHANGEVOLUME subcommand.
- Resetting the EDM attribute when a volume is purged via DFSMSrmm EDGTVEXT/EDGDFHSM programming interface.
- Considering the volumes with EDM attribute to be not eligible for DFSMSrmm expiration processing.

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 251 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 the specified number of data sets. Otherwise, all the data sets that have met the search criteria are returned.

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

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.

The ISPF dialog uses DFSMSrmm TSO subcommands to add and update resources. To process the TSO subcommands, you need to be authorized. See *z/OS DFSMSrmm Implementation and Customization Guide* for more details about authorizing DFSMSrmm users and ensuring security.

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 10, "Using RMM TSO subcommands," on page 187 for more information on using TSO subcommands.

Defining shelf locations

In your installation, you store your tape volumes and other removable media in shelf locations defined to DFSMSrmm by rack and bin numbers. Rack numbers 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.

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 208 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:

© Copyright IBM Corp. 1992, 2022

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 311 or “DELETEBIN: Deleting bin number information” on page 305 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 routine EDGUX100 for exit EDG_EXIT100 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 211 for the volume information DFSMSrmm obtains from the TCDB.

If you use the LABEL SCAN function of the EDGINERS utility to determine the contents of a volume that is not defined to DFSMSrmm, you can use the details for the first file to add the volume and first file data set information to the DFSMSrmm CDS. See “Initializing, erasing, and scanning tape volumes manually” on page 172 for information on scanning volume labels.

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 manually define a volume that is a start of a volume set, you can specify the retention method for the volume set. If you do not specify a retention method, DFSMSrmm will use the default retention method specified by the RETENTIONMETHOD option and its suboptions in EDGRMMxx.

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 type and recording format for volumes. Table 5 on page 28 shows the correspondence between the DFSMSrmm media types and the media types used in the system-managed tape data class construct.

Table 5. DFSMSrmm media types

The media type	Corresponds to the DFSMSrmm media type
MEDIA1	CST
MEDIA2	ECCST
MEDIA3	HPCT
MEDIA4	EHPCT
MEDIA5	ETC
MEDIA6	EWTC
MEDIA7	EETC
MEDIA8	EEWTC
MEDIA9	EXTC
MEDIA10	EXWTC
MEDIA11	EATC
MEDIA12	EAWTC
MEDIA13	EAETC

Table 6 on page 28 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 recording format set by DFSMSrmm when you do not specify a value, or when you specify a value that is invalid for the specified media type and is lower than the default recording format. When the specified recording format is invalid for the specified media type, and higher than the actual default recording format, the highest valid recording format value is assigned instead.

As shown in Table 6 on page 28 , 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, EFMT2, EEFMT2, EFMT3, or EEFMT3 recording formats. If you specify media types EXTC/MEDIA9 or EXWTC/MEDIA10, you can only specify EFMT2, EEFMT2, EFMT3, EEFMT3 , EFMT4, or EEFMT4 recording formats.If you specify media types EATC/MEDIA11, or EAWTC/MEDIA12, or EAATC/MEDIA13, you can only specify EFMT4, or EEFMT4 recording formats.

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

Table 6. How DFSMSrmm assigns media type and recording format (continued)

Media type	Recording format	Default recording format
HPCT	128TRACK 256TRACK 384TRACK	128TRACK
EHPCT	128TRACK 256TRACK 384TRACK	128TRACK
ETC/MEDIA5	EFMT1 EFMT2 EEFMT2 EFMT3 EEFMT3	EFMT1
EWTC/MEDIA6	EFMT1 EFMT2 EEFMT2 EFMT3 EEFMT3	EFMT1
EETC/MEDIA7	EFMT1 EFMT2 EEFMT2 EFMT3 EEFMT3	EFMT1
EEWTC/MEDIA8	EFMT1 EFMT2 EEFMT2 EFMT3 EEFMT3	EFMT1
EXTC/MEDIA9	EFMT2 EEFMT2 EFMT3 EEFMT3 EFMT4 EEFMT4	EFMT2
EXWTC/MEDIA10	EFMT2 EEFMT2 EFMT3 EEFMT3 EFMT4 EEFMT4	EFMT2
EATC/MEDIA11	EFMT4 EEFMT4	EFMT4
EAWTC/MEDIA12	EFMT4 EEFMT4	EFMT4
EAETC/MEDIA13	EFMT4 EEFMT4	EFMT4

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 211 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. If you want the volume retention to be expiration date managed (and not VRS managed), then specify EXPDT for retention method. 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 211 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 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, or the F line command from Volume Search results list, or from the Volume Release 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 sample exit module EDGUX100 for installation exit EDG_EXIT100, 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 EDG_EXIT100 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 269 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 32 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 on page 32 . Enter Y to update all volumes in a multivolume set.

```
EDGPT330
  Do you want to process all volumes in the multivolume 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. RETENTIONMETHOD and RETAINBY can be changed only for the first volume in the set and is propagated to all other volumes in the 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.

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 with 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 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. These operands contain non-restricted volume information:

ACCESS
ACCOUNT
ADDUSERS
CONFIRMRELEASE
DELUSERS
DESCRIPTION
EXPDT
EXPTM
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. Here are the operands you can change if you are authorized to use the CHANGEVOLUME subcommand FORCE operand:

ASDATE
ASTIME
COMPACTION
DENSITY
DSNAME
JOBNAME
LABEL
MEDIANAME
MEDIATYPE
MEDINF
NEWVOLUME
ORIGINALEXPDT
READDATE
RECORDINGFORMAT
WRITEDATE

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.

Changing volume information based on volume status

These operands can be used only if the volume is in master or user status.

ACCESS
ADDUSERS
COMPACTION
DELUSERS
DESCRIPTION
DSNAME
EDM/NOEDM
EXPDT
EXPTM
FEATCD
JOBNAME
LEVEL
LOANLOC
NUMBER
OWNER
OWNERACCESS
PREVVOL
READDATE
RELEASEACTION
RETPD
SECLEVEL
STORGRP
VOL1
WRITEDATE

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).

You can use these operands for all volumes regardless of their status:

ASDATE
ASTIME
CRDATE
CRSYSID
CRTIME
DENSITY
EJECT
HOME
INIT
LABEL
LOCATION
MEDIATYPE
POOL
RACK
RECORDINGFORMAT
REPLACE
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.

Deleting volume information

DFSMSrmm deletes volume information from the control data set when you release a volume. See “Releasing volumes” on page 113 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 32 . Enter Y to delete or release all volumes in a multivolume set.
3. DFSMSrmm displays the DFSMSrmm Processing Volume Set panel, which contains a list of all the volumes in the multivolume 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.

Reclaiming volumes from pending status or scratch status

To reclaim volumes:

- From pending release, use the CHANGEVOLUME subcommand with a volume serial number and either the RETPD or EXPDT operands. You can also add the HOLD operand to prevent automatic expiration and also prevent the use of the RMM DELETEVOLUME subcommand with the RELEASE operand.
- From scratch status and recover the data on the volume, use the CHANGEVOLUME subcommand with either the STATUS(USER) or STATUS(MASTER) operands. You can also add the HOLD operand to prevent automatic expiration and also prevent the use of the RMM DELETEVOLUME subcommand with the RELEASE operand. 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 these operands:

ASDATE
ASTIME
AUTOMOVE
BIN
CONFIRMMOVE
CONTAINER
CRDATE
CRSYSID
CRTIME
EDM/NOEDM
EJECT
HOME
LOANLOC
LOCATION
MANUALMOVE
MEDIANAME
POOL
RACK
READDATE
RETAINBY
RETENTIONMETHOD
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 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 where you want a volume returned when it is no longer retained by a vital record specification. When you use the LOCATION operand to change a volume's

location and do not use HOME, the HOME location for the volume is set to the value specified for LOCATION.

Confirming volume moves and release actions

You must confirm volume movement and actions before DFSMSrmm can perform additional movement or actions you require. For example, 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 by 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 95 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 specifying an 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 327 for more information.

Use CHANGEVOLUME with the NOCONFIRMMOVE or the NOCONFIRMRELEASE operand and a volume serial number of * to undo a global confirmation that changes the status of a move or action from CONFIRMED to PENDING.

Use the READYTOSCRATCH operand to confirm moves for volumes that are ready to return to scratch. Use the NOTREADYTOSCRATCH operand to confirm moves for volumes that are private volumes or for volumes with release actions other than scratch pending.

Overriding automatic move processing

Use the CHANGEVOLUME subcommand with the MANUALMOVE operand to manually control the movement of volumes. Use the CHANGEVOLUME subcommand with the AUTOMOVE operand to return to automatic processing. See "Using manual move control" on page 90.

To override automatic movement, you can put a volume under manual move control by 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 to change 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 or reuse

Because physical 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. Logical WORM tapes can be returned to the scratch pool, allowing the volumes to be reused either as WORM or R/W logical volumes. You can use the volume release actions to control how DFSMSrmm manages this process.

By default, DFSMSrmm sets the volume release action as follows:

1. For physical WORM, the return to owner release action is set, but you can use the replace release action instead, by changing the release action at any time during the volume's life.

   DFSMSrmm REPLACE policy implementation for physical WORM tapes results in the REPLACE release action for tapes that meet the replacement criteria. IBM recommends that you use the return to owner release action, so that any WORM volume with the replace release action is known to need replacement based on the REPLACE policy.
2. For logical WORM, the replace release action is set.

You can use the release actions for the volumes to determine the processing required. For physical WORM:

- You can 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) or RMM CHANGEVOLUME *volser* CRLSE(RETURN) command to confirm the release action, which deletes the volume information from the DFSMSrmm control data set.
- You can 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(REPLACE) or RMM CHANGEVOLUME *volser* CRLSE(REPLACE) command to confirm the release action, which resets the volume information to reflect that the volume has been replaced. Alternatively, if you do not want to replace the volumes, but would rather delete them from the DFSMSrmm control data set, use the RMM DELETEVOLUME *volser* REPLACE subcommand.
- When the REPLACE release action is set based on the REPLACE policies, but is not yet pending, you should consider whether to copy the data to a newer WORM volume and destroy the WORM volume that has now matched your replacement policies. Use the DFSMSrmm VRS retention information for the data sets on the volume and the volume expiration date to determine how much longer the data needs to be retained.

You do not need to take any action to have logical volumes (whether WORM or R/W) automatically returned to scratch and available for reuse as either logical WORM or as R/W logical volumes.

To prevent logical WORM volumes returning to scratch automatically, use VLPOOL with RELEASEACTION(NOTIFY). This forces the NOTIFY release action for all volumes in the pool and requires you to issue RMM CV volser CRLSE(NOTIFY) to confirm the action before the REPLACE is handled automatically. Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for information about how to use NOTIFY and ensure that it requires a manual action.

Changing the status of WORM tapes

DFSMSrmm prevents physical 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 after it has been used for input or output. You can change the status of a WORM tape to scratch status using one of these two techniques:

1. Use the RMM DELETEVOLUME *volser* FORCE command to delete the volume. Next, use the RMM ADDVOLUME *volser* STATUS(SCRATCH) command to add the volume to DFSMSrmm with the scratch status.
2. Use the RMM CHANGEVOLUME *volser* RELEASE(REPLACE) RETPD(1) command, followed by the RMM DELETEVOLUME *volser* RELEASE command, followed by the RMM CHANGEVOLUME *volser* 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. Optionally, use the RMM CHANGEVOLUME *volser* CONFIRMRELEASE(INIT) command to confirm that the volume is already labeled. Next, run DFSMSrmm expiration processing to set the volume to scratch status.

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, or a valid e-mail address, 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. These topics 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 202 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, or a valid e-mail address.

   Note: 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 265 for more information.

Restriction: You can change information only 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

Note: Use the DELETEOWNER subcommand to delete an owner ID from DFSMSrmm. See "DELETEOWNER: Deleting owner information" on page 309 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 topic 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 205 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 27 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 268 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 142 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 310 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 and time, which is the date and time that the data set can be considered for deletion.
- LASTREF extra days, or NOLASTREF, if the data set is on a volume managed by the EXPDT retention method.
- 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 EDG_EXIT100 to ensure that DFSMSrmm records information only 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 192 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 251 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 307 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. Retention methods

Among the important decisions to be made when using DFSMSrmm are how to retain tape data sets and for how long. You might want to retain a given data set for a specific period of time after it is created, or retain it based on some event (for example, while the data set is cataloged or when it was last used), or retain it permanently.

DFSMSrmm provides two retention methods for retaining tape data sets:

- The EXPDT retention method, which allows expiration to be on a specific date and time or based on an event (catalog retention or retention for a specific number of days after last use).
- The VRSEL retention method, which uses vital record specifications to implement retention and movement policies through which a retention date is calculated each time that VRSEL inventory management processing is run. DFSMSrmm retains a volume based on this retention date and on the volume expiration date and time.

The retention method is an attribute of the volume. All volumes in a multivolume set have the same retention method. All data sets on a volume are managed with the same retention method as the volume on which they reside. DFSMSrmm will assign a retention method to a new single volume or new multivolume set when the first tape data set of the single volume or first volume of the multivolume set is created. The retention method can be set in the:

1. Default retention method that is specified in the DFSMSrmm parmlib option.
2. DFSMSrmm installation exit EDG_EXIT100, if the (generic) data set name is specified.
3. RMM ADDVOLUME command using the RETENTIONMETHOD parameter.
4. Retention method can be specified in the SMS Management Class.
5. Retention method can be specified in the defaults table, using the EDGDEFxx PARMLIB member.

Retention of multivolume sets

The choice of retention method affects how multivolume sets are retained:

- With the EXPDT retention method, volumes and volume sets can be retained as individual volumes, as volume sets, or based on the expiration date of the first file. The volume attribute related to the retention of a multivolume set is the RETAINBY attribute. All volumes in a multivolume set have the same RETAINBY value (VOLUME or SET or FIRSTFILE). DFSMSrmm will assign a RETAINBY value to a new retention method EXPDT single volume or multivolume set when the first tape data set is created. The RETAINBY attribute can be set from:
  - "Volume Set Management Level" in the SMS Management Class.
  - The defaults table, defined in the EDGDEFxx PARMLIB member.
  - parmlib OPTION RETENTIONMETHOD(EXPDT(RETAINBY(value))), as described in *z/OS DFSMSrmm Implementation and Customization Guide*.
  - RMM ADDVOLUME command using RETAINBY operand, as described in “ADDVOLUME: Adding volume information” on page 211.
- With the VRSEL retention method, multivolume sets can be retained as individual volumes or as volume sets. The parmlib OPTION RETAINBY is the system wide setting for all multivolume sets. The parmlib OPTION RETAINBY is described in *z/OS DFSMSrmm Implementation and Customization Guide*.

The expiration date

One important parameter for both retention methods is the expiration date of the data set and the volume. The data set expiration date or retention period is determined when a new tape data set is created. The expiration date or retention period can be specified at multiple levels:

© Copyright IBM Corp. 1992, 2022

1. The default retention period RETPD specified in the DFSMSrmm parmlib as a global default.
2. Separate defaults specific to GDG or non-GDG data sets managed by the EXPDT retention method specified in the DFSMSrmm parmlib.
3. The retention period specified in the defaults table, using the EDGDEFxx PARMLIB member.
4. The expiration attribute "Expire after Date/Days" in the DFSMS management class, if the use is enabled by the DFSMSrmm MCATTR parmlib option and if the data set is associated with such a management class. For information on how the exploitation of management class expiration attributes can be enabled in DFSMSrmm see “Using the parmlib member EDGRMMxx” in *z/OS DFSMSrmm Implementation and Customization Guide*. For information on exploiting the management class attributes for tape, see “Exploiting Management Class attributes” *z/OS DFSMSrmm Implementation and Customization Guide*
5. The EXPDT or RETPD in the DFSMS data class, if the data set is associated with a DFSMS data class
6. The JCL DD statement, using the EXPDT or RETPD keywords
7. The DFSMSrmm installation exit EDG_EXIT100. For more details about the use of the special dates in exit module EDGUX100, see "Assigning Expiration Dates" in *z/OS DFSMSrmm Implementation and Customization Guide*.

This list is in reverse priority order. This means, if the expiration or retention is specified at multiple levels, then the later one takes precedence. However, it is possible to override the JCL DD statement in the defaults table.

When a volume is added with the RMM GETVOLUME command, the expiration date from the volume catalog is used, unless EXPDT or RETPD is supplied with the command.

The expiration date cannot exceed the maximum retention period MAXRETPD set by your installation in the DFSMSrmm parmlib member. If the expiration date exceeds MAXRETPD then the expiration date will be replaced with the current date plus the maximum retention period.

If a data set is extended with DISP=MOD, DFSMSrmm ensures that the expiration date cannot be lowered, which means that the maximum of the current expiration date and the newly supplied expiration date is taken. Management class expiration attributes are not retrieved at job steps with DISP=MOD.

The expiration date of the data sets influences the expiration date of the volumes. The retention method used for a volume determines how DFSMSrmm processes the data set and volume expiration date:

- With the EXPDT retention method:
  - For volumes retained by VOLUME or SET, the processing is as follows:

    When creating a data set, DFSMSrmm processing will determine the highest data set expiration date and time of all data sets on the volume and, if it is higher than the volume expiration date and time , the volume expiration date and time will be changed to the highest expiration date and time. Also, when the data set expiration date and time is changed to a higher value using the RMM CHANGEDATASET command, then the volume expiration date and time is set to the highest date and time. The volume expiration date and time cannot be lowered with the RMM CHANGEDATASET command. However, you can use the RMM CHANGEVOLUME command to change the volume's expiration date to a lower date. DFSMSrmm EXPROC inventory management only uses the volume expiration date for release processing.

    All files of the same multivolume data set have the same expiration date and time. DFSMSrmm synchronizes the data set records while tape data sets are processed and when an RMM ADDDATASET, ADDVOLUME, CHANGEDATASET, or CHANGEVOLUME command is issued. When you use the RMM CHANGEVOLUME RETENTIONMETHOD(EXPDT) command to convert a multivolume set from the VRSEL retention method to the EXPDT retention method, DFSMSrmm performs additional processing for all multivolume data sets in the volume set: The maximum expiration date and time for each multivolume data set is determined from the data set records and used to equalize the expiration date for the multivolume data set. In addition, for volumes retained by SET, the highest volume expiration date and time is propagated to all volumes in the set, the RMM CHANGEVOLUME EXPDT() command will update the expiration dates of all volumes in the set to the specified date.
  - For volumes retained by FIRSTFILE, the processing is as follows:

The expiration date and time of the first file is used to determine the expiration date and time of the volume or multivolume set. All volumes in a set will have the exact same expiration date. DFSMSrmm does not perform any additional processing to equalize the expiration date or LASTREF extra days in a multivolume data set

- With the VRSEL retention method, when creating a data set, DFSMSrmm processing will determine the highest data set expiration date and time of all data sets on the volume and, if it is higher than the volume expiration date and time, the volume expiration date and time will be changed to the highest expiration date and time. Also, when the data set expiration date or time is changed to a higher value using the RMM CHANGEDATASET command, then the volume expiration date and time is set to the highest date and time. The volume expiration date cannot be lowered with the RMM CHANGEDATASET command. However, you can use the RMM CHANGEVOLUME command to change the volume's expiration date or time to a lower date or time. Every file of a multivolume data set has its own expiration date and time. Changing the expiration date on one file of the multivolume data set has no effect on the other files of the multivolume data set.

DFSMSrmm EXPROC inventory management only uses the volume expiration date for release processing.

Note: Expiration dates derived from Model DSCB (DCB=model.dsname) or LIKE=dsname are not evaluated as KEYDATE by EDGUX100, DEFTABLE or SMS Management Class ACS routine. Only EXPDT=yyddd specified in JCL, DATACLAS or dynamic allocation are evaluated as KEYDATE. However, EXPDT derived from Model DSCB will be stored in DFRMM Volume and Data set records and will be used as valid expiration dates if no further matches occur from DEFTABLE or VRS.

Expiration based on last reference

You can control the expiration date of data sets based on their last reference date.

- With the EXPDT retention method, the expiration date can be set based on the last usage. DFSMSrmm uses the data set LASTREF extra days to evaluate the data set expiration date and time. The extra days are added to the date of last reference. The data set expiration date is set to the maximum of the dates calculated by using data set LASTREF extra days and the current expiration date. Any reference to the data set by a write or read operation will redetermine the expiration date. When a new tape data set is created the LASTREF extra days attribute can be set from:
  - The parmlib OPTION RETENTIONMETHOD(EXPDT(LASTREF(extra_days)))
  - The LASTREF parameter of the defaults table specified in the EDGDEFxx parmlib member
  - The management class expiration attribute 'Expire after Days-Non-Usage', if the use is enabled by DFSMSrmm MCATTR parmlib option and if the data set is associated with such a management class. For information on how the exploitation of management class expiration attributes can be enabled in DFSMSrmm see "Using the parmlib member EDGRMMxx" in *z/OS DFSMSrmm Implementation and Customization Guide*. For information on exploiting the management class attributes for tape, see "Exploiting Management Class attributes" *z/OS DFSMSrmm Implementation and Customization Guide*
  - TSO command ADDDATASET / CHANGEDATASET LASTREF(extra_days)
- With the VRSEL retention method, you can specify the LASTREFERENCEDAYS operand when adding a data set or name vital record specification for data sets to be retained based on last usage. For more information, see "ADDVRS: Adding a vital record specification" on page 236.

Summary of the tasks to perform

Perform the following tasks:

- Use the RETENTIONMETHOD parmlib option if you want to set a default system-wide retention method. If you do not specify a default retention method, the system uses the VRSEL retention method as default. Set system wide parmlib OPTIONs: default retention period RETPD, the maximal retention period MAXRETPD, and CATRETPD, EXPDTDROP, MOVEBY, RETAINBY, VRSDROP, VRSJOBNAME, VRSCHANGE, VRSMIN, and VRSRETAIN in the parmlib member EDGRMMxx. For the EXPDT retention method, specify OPTION RETENTIONMETHOD(EXPDT(RETAINBY() LASTREF())) in the parmlib member

EDGRMMxx. See *z/OS DFSMSrmm Implementation and Customization Guide* for information on the OPTION RETENTIONMETHOD parmlib operand.

- Optionally update the defaults table or EDG_EXIT100 installation exit to set the retention method to be used for new tape data. See *z/OS DFSMSrmm Implementation and Customization Guide* for information on the EDG_EXIT100 installation exit.
- You can set the retention method of a new volume and the RETAINBY value when adding it to DFSMSrmm by using the RMM ADDVOLUME RETENTIONMETHOD command, as described in "ADDVOLUME: Adding volume information" on page 211.
- If you want to change the retention method and the RETAINBY value of a master or user volume or multivolume set use the RMM CHANGEVOLUME RETENTIONMETHOD against the volume or the first volume of the multivolume set. All volumes in the multivolume set will automatically get the same retention method. See "CHANGEVOLUME: Changing volume information" on page 269 for more information.

Deciding between the EXPDT and VRSEL retention methods

There are several things to be considered when deciding which retention method is the best for your volumes, multivolume sets, and the data sets on these volumes.

Considerations for data sets and volumes and volume sets subject to the EXPDT retention method:

- All data sets on a volume or volume set are excluded from DFSMSrmm VRSEL inventory management processing. Only EXPROC processing is required to handle expiration of volumes managed by the EXPDT retention method.
- When volumes are changed from being managed by the VRSEL retention method to be managed by the EXPDT retention method, you help reduce the processing that DFSMSrmm VRSEL inventory management must perform.
- Volume sets managed by the EXPDT retention method can be retained by VOLUME, by SET, or by FIRSTFILE. The corresponding parmlib OPTION is RETENTIONMETHOD(EXPDT(RETAINBY())).

  Note: The parmlib OPTIONs RETAINBY and MOVEBY apply to the VRSEL retention method only.
- DFSMSrmm maintains a consistent expiration date and time for all data set records of a multivolume data set, except for multivolume data sets on volumes retained by FIRSTFILE.
- Data sets can be retained based on their usage. The data set LASTREF extra days attribute causes the data set expiration date to be recalculated when the data set is used for a read or write operation.
- Volumes containing data sets that are closed by ABEND processing or marked deleted or open are handled as if no special ABEND/DELETED/OPEN VRS had been defined. As they are not considered for the VRSEL processing, no VRS can be assigned.
- Volumes managed by the EXPDT retention method are included only in the EXPDTDROP limit. VRSRETAIN and VRSDROP limits apply only to volumes managed by VRSEL retention method.
- Return to scratch is attempted in a single run of inventory management EXPROC processing. (Similar as if the RELEASE(SCRATCHIMMEDIATE) release option had been used for volumes managed by the VRSEL retention method).
- The EXPDT retention method does not control movement, so if you want to move a volume retained by the EXPDT retention method, you must move it manually. If you require DFSMSrmm to automatically manage the movement of a volume, use the VRSEL retention method instead.
- You can specify that volumes purged from DFSMShsm are to be retained a few extra days to be sure that purged volumes do not contain any data that might still be needed. DFSMShsm migration and backup volumes can be held by RMM using the EDM option, and an expiration date can be set to the current or future date at the time the volume is purged from DFSMShsm with the EDGTVEXT call. See the parmlib options EDM and TVEXTPURGE with the EXPIRE(days) subparameter.
- Changes to vital record specifications will not affect data sets and volumes managed by the EXPDT retention method.
- Any time you know when your volume is going to expire, just by looking at the expiration date.

- EXPROC processing provides a summary of volumes by retention method.

Considerations for data sets and volumes subject to VRSEL retention method:

- The VRSEL retention method uses vital record specifications to set policies for retention and automatic movement of your removable media. You can tailor you retention and movement policies in the vital record specifications according your needs.
- DFSMSrmm VRSEL inventory management processing is needed to process the retention and movement policies specified in vital record specifications.
- By excluding one or more of the data sets on your volume from VRSEL (VRSELEXCLUDE), you can have the volume managed just by those data sets not excluded from VRSEL. See CHANGEDATASET command. See the topic "Excluding data sets from VRSEL processing" in *z/OS DFSMSrmm Implementation and Customization Guide* for more information.
- When a volume is dropped from retention (is not retained by a VRSEL any more) or if the data set or volume does not match to any vital record specifications, or your volume is not retained by set, then your volume is eligible for expiration processing
- For detailed information on the VRSEL retention method, see:
  - Chapter 4, “Defining vital record specifications,” on page 53
  - Chapter 5, “Using vital record specifications to retain and move volumes,” on page 73
  - Chapter 6, “Using vital record specifications to define retention and movement policies,” on page 99

  Also see the topic in "How Expiration Processing Works " in *z/OS DFSMSrmm Implementation and Customization Guide*.

Candidates for exploitation of the EXPDT retention method

- Any volume that is externally managed by an application such as DFSMShsm, DFSMS OAM, or Tivoli® Storage Manager that is assigned an expiration date such as 1999/365. When the volume is no longer required, the application exploits the EDGTVEXT program interface to expire or release the volume.
- "Archive" tape data that, once created, is retained by the same, simple policy until it should be expired, such as by using a retention period.
- Any volume currently managed using a data set or volume VRS with DAYS COUNT(nnnn).
- Any volume currently managed using a VRS with UNTILEXPIRED causing retention by the volume EXPDT.
- Any volume currently managed using a data set VRS or a name VRS with LASTREFERENCEDAYS COUNT(nnnn).
- Any volume that can be managed based on the expiration date of the first file.
- Any volume that is to be retained as long as any of its data sets are cataloged.

Changing existing DFSMShsm managed volumes: You can use the CHANGEVOLUME subcommand with RETENTIONMETHOD and RETAINBY operands on single volumes and the first volumes in a multivolume set. DFSMSrmm will apply the specified retention method to all volumes in the multivolume set. All EXPDT retention method managed volumes must have a valid expiration date. For multivolume sets that use the EXPDT retention method and retain by VOLUME the desired expiration date must be specified for each volume in a multivolume set.

1. Identify your DFSMShsm managed single volumes, build and execute CLIST command for changing retention method, RETAINBY value, and expiration date:

```
RMM SV OWNER(hsm) INSET(NO) LIMIT(*) ... CLIST('RMM CV ',' RM(EXPDT) RETAINBY(VOLUME) EXPDT(99365)')
EXEC EXEC.RMM.CLIST
```

2. Identify your DFSMShsm managed first volumes in multivolume sets, build and execute CLIST command for changing retention method:

```
RMM SV OWNER(hsm) INSET(FIRST) LIMIT(*) ... CLIST('RMM CV ',' RM(EXPDT)')
EXEC EXEC.RMM.CLIST
```

3. Identify your DFSMShsm managed multivolume sets, build and execute CLIST command for changing expiration date:

```
RMM SV OWNER(hsm) INSET(YES) LIMIT(*) ... CLIST('RMM CV ',' EXPDT(99365)')
EXEC EXEC.RMM.CLIST
```

For single volumes and first volumes in each multivolume set that you want to retain by SET, issue the following:

1.
```
RMM SV OWNER(id) INSET(NO) LIMIT(*) ... CLIST('RMM CV ',' RM(EXPDT) RETAINBY(SET) EXPDT(99365)')
EXEC EXEC.RMM.CLIST
```

2.
```
RMM SV OWNER(id) INSET(FIRST) LIMIT(*) ... CLIST('RMM CV ',' RM(EXPDT) RETAINBY(SET) EXPDT(99365) ')
EXEC EXEC.RMM.CLIST
```

DFSMSrmm processing sets the expiration date for all volumes in the set to the expiration date specified for the first volume.

For single volumes and first volumes in each multivolume set that you want to retain by FIRSTFILE, issue the following :

1.
```
RMM SV OWNER(id) INSET(NO) LIMIT(*) ... CLIST('RMM CV ',' RM(EXPDT) RETAINBY(FIRSTFILE) ')
EXEC EXEC.RMM.CLIST
```

2.
```
RMM SV OWNER(id) INSET(FIRST) LIMIT(*) ... CLIST('RMM CV ',' RM(EXPDT) RETAINBY(FIRSTFILE) ')
EXEC EXEC.RMM.CLIST
```

You cannot set the volume expiration date in this case, the expiration date from the first file is applied. You can change the expiration date using RMM CHANGEDATASET command on the first file. DFSMSrmm processing sets the expiration date for all volumes in the set to the expiration date specified for the first file.

Changing retention method to EXPDT for existing VRS retained data

For single volumes and first volumes in each multivolume set that you want to retain by VOLUME, issue the following subcommand:

```
RMM CV volser RM(EXPDT) RETAINBY(VOLUME) EXPDT(retdate)
```

You must also set the desired expiration date for all other volumes in each multivolume set using the following subcommand:

```
RMM CV volser EXPDT(retdate)
```

For single volumes and first volumes in each multivolume set that you want to retain by SET, issue the following subcommand:

```
RMM CV volser RM(EXPDT) RETAINBY(SET) EXPDT(retdate)
```

DFSMSrmm processing sets the expiration date for all volumes in the set to the expiration date specified for the first volume.

For single volumes and first volumes in each multivolume set that you want to retain by FIRSTFILE, issue the following subcommands:

```
RMM CV volser RM(EXPDT) RETAINBY(FIRSTFILE)
RMM CD dsname VOLUME(volser) SEQ(1) EXPDT(retdate)
```

You cannot set the volume expiration date in this case, the expiration date from the first file is applied. You can change the expiration date using RMM CHANGEDATASET command on the first file. DFSMSrmm processing sets the expiration date for all volumes in the set to the expiration date specified for the first file.

Note: Do not specify RETPD(nn) because DFSMSrmm subcommand processing applies this to the current date and not the creation date. Instead use the volume retention date which is the date calculated by VRSEL.

For new data sets added to a volume changed from the VRSEL to the EXPDT retention method, a RETPD or EXPDT must be specified at the time of allocation or the EDG_EXIT100 installation exit must assign a date, or the default retention period will be applied.

For data sets that were previously retained by a VRS with LASTREFERENCEDAYS, you can obtain the same result with the EXPDT retention method by specifying for each data set:

```
RMM CD dsname VOLUME(volser) SEQ(n) LASTREF(extra_days)
```

Chapter 4. Defining vital record specifications

Review the information in this topic if you have decided to use the VRSEL retention method for any of your volumes and data sets. Vital record specifications are used only for retention and movement of volumes that are managed by the VRSEL retention method and data sets on these volumes that are not excluded from VRSEL processing.

Use vital record specifications to set policies for retaining and moving all removable media managed by the VRSEL retention method. 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 completed before any new ones can take effect or the volume can be released.

You can control the way DFSMSrmm retains and moves volumes that are managed by the VRSEL retention method and data set on these volumes that are not excluded from VRSEL processing 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 104 for information about defining the minimum vital record specification.
- Use the DFSMSrmm parmlib OPTION VRSDROP to establish your normal expectations for numbers or percentage of volumes being removed from VRS retention.
- Use the DFSMSrmm parmlib OPTION VRSRETAIN to establish your normal expectations of the numbers or percentages of newly assigned volumes to be retained by vital record specifications.
- Use the DFSMSrmm parmlib OPTION EXPDTDROP to establish your normal expectations for the numbers or percentages of volumes dropped from expiration date retention.
- 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 79.
- See “What vital records selection processing you can specify” on page 100 for more information about vital record processing.

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.

© Copyright IBM Corp. 1992, 2022

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 in order to match to data set names, data set name masks, management class names, vital record specification management values, and optionally, match to job names. A data set name vital record specification can also specify special retention for data sets that are:

- closed by abend processing (reserved name ABEND),
- created with normal disposition of DELETE (reserved name DELETED), or
- left open (reserved name OPEN).

In this way, 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 and job name masks 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.

If the data set is the subject of special ABEND, DELETED, or OPEN retention then the secondary vital record specification will also be the subject of special ABEND, DELETED, or OPEN retention, respectively.

See “Identifying data sets for retention” on page 55 for more information about defining data set vital record specifications.

Also see “Separating the data set name filter from the policy itself” on page 102 on the advantages of using separate VRSs for defining the data set name filter and the associated retention and movement information.

Name vital record specifications

Name vital record specifications can define additional move criteria and additional retention criteria. You can define name vital record specifications that can be chained to multiple data set or volume vital record specifications. You can define either retention or movement information, or both. When a name vital record specification specifies only move criteria, the name vital record specification can only define retention criteria that does not override the maximum retention value set by the previous vital record specification that contains retention criteria.

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 68 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. A subchain includes all the vital record specifications chained from the start of the subchain until the next subchain starts. 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 on page 55 andFigure 15 on page 55 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 on page 55 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 55 uses the name vital record specification N1 shown in Figure 15 on page 55 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 on page 55, 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 EDG2230I, retains any matching data sets, and sets a return code of 4.

See “Chaining retention and movement policies” on page 101 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, job name masks, management classes, and vital record specification management values. You can manage generation data group (gdg) data sets by using gdg base names or data set name masks. You can manage 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. Only data sets on volumes that are managed by the VRSEL retention method and that are not excluded from VRSEL processing are considered. DFSMSrmm compares the data set name and job 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: for gdg and pseudo gdg, the base name is used.

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 63 for additional information.

After inventory management vital record processing has completed, you can use the DFSMSrmm dialog to determine how a data set is being retained. Using the dialog, search for the data set and then use

the M line command (or the equivalent MATCHVRS primary command) to display the data set and view the matching VRS information. These commands use the matching VRS details, if any, from the data set information to list the matching VRSs. The primary and secondary VRSs are listed, if they exist. If there are no matching VRS details, dialog processing uses the fully qualified data set name to search for a matching VRS. In addition, the specific volume VRS is also listed, if it exists. You can then place the cursor on a matching VRS and press Enter to see what retention policies it specifies. See "Using point-and-shoot fields" on page 9 for information on the MATCHVRS primary command and its related point and shoot functions.

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)
```

DFSMSrmm does not check quoted data set names for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. Data set names without quotes must pass these data set naming rules:

- A data set name can have one or more qualifiers.
- Each qualifier is one to eight characters. The first character alphabetic (A to Z) or national (# @ $). The remaining seven characters can be either alphabetic, numeric (0 - 9), national, or a hyphen (-).
- Qualifiers are separated by a period.
- DFSMSrmm adds your TSO PROFILE PREFIX value as the high-level qualifier.
- The data set name must not include a member name.

When an existing record in the DFSMSrmm control data set is being processed, these items are checked:

- The name is 1 to 44 characters and enclosed in quotes if any special characters are included.
- If the data set name is not enclosed in quotes, PROFILE PREFIX is applied. There is no check against data set naming or data set mask naming rules.

DFSMSrmm uses single qualifier data set name masks to match to management class names, vital record specification management value names, and also the reserved name values of OPEN, DELETED, and ABEND. See "Special ABEND, DELETED, and OPEN retention" on page 59 for more information about these reserved name values. See "Management class and management value names" on page 59 for more information about using management class names or vital record specification management value names.

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 that you define.

DFSMSrmm does not check quoted data set name masks for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. In addition, for quoted data set name masks, DFSMSrmm calls DFSMSdfp common filter services (CFS) to ensure that these basic mask rules are met:

Period (.)
A leading or trailing period is not allowed. Consecutive periods are also not allowed.

Double asterisk (**)
Cannot be specified within a data set name qualifier.

Cannot be specified before a qualifier with a single asterisk. For example, specifying DSNAME(**.*) is not a valid data set name mask.

Data set name masks without quotation marks must pass these data set naming rules:

- A data set name can have one or more qualifiers.
- Each qualifier is one to eight characters. The first character alphabetic (A to Z) or national (# @ $). The remaining seven characters can be either alphabetic, numeric (0 - 9), national, or a hyphen (-).
- Qualifiers are separated by a period.
- DFSMSrmm adds your TSO PROFILE PREFIX value as the high-level qualifier.
- The data set name must not include a member name.

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 if 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.

This 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 67 for more information about identifying pseudo-GDGs for retention and movement.

When an existing record in the DFSMSrmm control data set is being processed, these items are checked:

- The name is 1 to 44 characters and enclosed in quotation marks if any special characters are included.

- If the data set name is not enclosed in quotations, PROFILE PREFIX is applied. There is no check against data set naming or data set mask naming rules.

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, this vital record specification could be defined.

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

DFSMSrmm uses the reserved job names of ABEND, DELETED, and OPEN. See “Special ABEND, DELETED, and OPEN retention” on page 59 for more information about ABEND, DELETED, and OPEN. Use JOBNAME(ABEND*), JOBNAME(DELETED*), or JOBNAME(OPEN*) to match to actual jobnames of ABEND, DELETED, or OPEN.

The VRSJOBNAME parmlib option allows you to specify how job names are used in the data set matching process. See “How DFSMSrmm uses jobname to apply policies” on page 62 for more information.

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 on page 59shows 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

Management class and management value names

You can specify a management class name or vital record specification management value in a data set name 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 z/OS data set naming conventions. This name must be a single qualifier, but can include mask characters to create a single generic qualifier covering multiple management classes or management values. To create a generic management class name or management value, use an * to match any character string and % to match any one character. For example, you can define the management class name MC123* for all data sets with a management class starting with the character MC123.

Special ABEND, DELETED, and OPEN retention

You can use the reserved data set or job names ABEND, DELETED, and OPEN to specify policies for data sets closed as a result of an abnormal end in a task, that were used or created with normal disposition DELETE, or for data sets that are left open or are in use during inventory management. This allows you to use either data set name masks or job name masks, but not both, to manage abended, deleted, or open data sets. The data set name mask or job name can even be used to match to data sets by management class or vital record specification management value, as long as the data set name mask specified is no more than a single qualifier. Thus, both a primary and a secondary matching vital record specification is possible. If a special ABEND, DELETED, or OPEN primary vital record specification is found, DFSMSrmm will look for a special ABEND, DELETED, or OPEN secondary vital specification, respectively. In the case where the primary and secondary matching vital record specifications found would be the same, DFSMSrmm will use them only as the primary vital record specification.

The matching order is: OPEN, then DELETED, then ABEND. If no matching vital record specification is found for special ABEND, DELETED, and OPEN retention, then DFSMSrmm looks for a vital record specification that matches the normal data set.

Only data sets on volumes that are managed by the VRSEL retention method and that are not excluded from VRSEL processing are considered for matching. If you are using an external data management application, such as DFSMShsm, see the topic "Running DFSMSrmm with DFSMShsm" in *z/OS DFSMSrmm Implementation and Customization Guide* for further recommendations.

ABEND support

You enable the management function by defining one or more ABEND VRSs. Abended data sets that do not match to a ABEND VRS are matched to other VRSs as normal.

```
RMM ADDVRS DSNAME('ABEND') LASTREFERENCEDAYS COUNT(5)
```

```
RMM ADDVRS DSNAME('**') JOBNAME(ABEND) DAYS COUNT(1)
```

When VRSEL processing finds a data set that is subject to ABEND processing, it first attempts to match the data set to the special ABEND vital record specifications and uses the exact same matching order and selection rules as it does for regular data sets. The matching scope is limited to those vital record specifications specified with ABEND job names and data set names. Thus, both a primary and a secondary matching vital record specification is possible, as long as they are not the same. If both vital record specifications found are the same, then they are used only as the primary vital record specification. If you

need to use the ABEND job names as non-special values, you have to use generic job names in the vital record specification, such as ABEND*.

DFSMSrmm implements the ABEND vital record specifications as follows:

Note: DFSMSrmm does just only one of the following, but in this sequence.

- If the DSNAME mask is 'ABEND', the data set name mask of '**' is used. The JOBNAME mask is used as it is.
- If the JOBNAME mask is 'ABEND', the job name mask of '*' is used. Also, no JOBNAME mask is used in case any of the data sets do not have a job name.

Thus, it is possible to define multiple vital record specifications that result in the same masks being used, or that may not be used as expected. For example,

- DSNAME('ABEND') without job name and DSNAME('**') JOBNAME(ABEND) are equivalent, but the latter is used by DFSMSrmm. EDG2219I issued for the former because of key sequence.
- DSNAME('ABEND') JOBNAME(*) and DSNAME('**') JOBNAME(ABEND) are equivalent, but the latter is used by DFSMSrmm filter matching processing. The former is ignored, because of key sequence. Matching data sets show that ABEND vital record specification has been used.
- DSNAME('ABEND') JOBNAME(ABEND) results in a mask of DSNAME('**') JOBNAME(ABEND) that only matches cases where the job name actually is ABEND'.

VRSEL processing issues message EDG2219I if any conflicting definitions are created.

DELETE support

DFSMSrmm provides support for managing deleted tape data sets. DFSMSrmm checks the normal disposition at CLOSE time and if this is 'DELETE', a 'deleted' flag is recorded in the data set record. Subsequent use of a data set cannot change this. However, you can use the CHANGEDATASET command to reset the flag.

The DELETE disposition does not have to be specified on the job step where the tape data set is created. It could be specified in a different step or even a different job. Tape data sets are subject to VRS processing as specified by your retention and movement policies. To ensure that 'deleted' data sets are managed differently, you must create special DELETED VRSs. Otherwise, they are managed by the normal matching VRS.

A VRS can use the DELETED value. 'DELETED' is a restricted VRS data set name and a restricted VRS JOBNAME.

DFSMSrmm implements the DELETED vital record specifications as follows:

Note: DFSMSrmm does just only one of the following, but in this sequence.

- If the DSNAME mask is 'DELETED', the data set name mask of '**' is used. The JOBNAME mask is used as it is.
- If the JOBNAME mask is 'DELETED', the job name mask of '*' is used. Also, no JOBNAME mask is used in case any of the data sets do not have a job name.

Thus, it is possible to define multiple vital record specifications that result in the same masks being used, or that may not be used as expected. For example,

- DSNAME('DELETED') without job name and DSNAME('**') JOBNAME(DELETED) are equivalent, but the latter is used by DFSMSrmm. EDG2219I issued for the former because of key sequence.
- DSNAME('DELETED') JOBNAME(*) and DSNAME('**') JOBNAME(DELETED) are equivalent, but the latter is used by DFSMSrmm filter matching processing. The former is ignored, because of key sequence. Matching data sets show that DELETED vital record specification have been used.
- DSNAME('DELETED') JOBNAME(DELETED) results in a mask of DSNAME('**') JOBNAME(DELETED) that only matches cases where the job name actually is 'DELETED'.

VRSEL processing issues message EDG2219I if any conflicting definitions are created.

You enable the management function by defining one or more DELETED VRSs. Both a primary and a secondary matching vital record specification is possible, as long as they are not the same. If both vital record specifications found are the same, then they are used only as the primary vital record specification. Deleted data sets that do not match to a DELETED VRS are matched to other vital record specifications, as normal.

```
RMM AS DSNAME('DELETED') DAYS COUNT(0) RELEASE(EXPIRYDATEIGNORE,SCRATCHIMMEDIATE)
RMM AS DSNAME(mask) JOBNAME(DELETED) DAYS COUNT(0) +
             RELEASE(EXPIRYDATEIGNORE,SCRATCHIMMEDIATE)
```

OPEN support

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

```
RMM ADDVRS DSNAME('MOWI.**') JOBNAME(OPEN) DAYS COUNT(99)
```

When VRSEL processing finds a data set that is subject to OPEN processing, it first attempts to match the data set to the special OPEN vital record specifications and uses the exact same matching order and selection rules as it does for regular data sets. The matching scope is limited to those vital record specifications specified with OPEN job names and data set names. Thus, both a primary and a secondary matching vital record specification is possible , as long as they are not the same. If both vital record specifications found are the same, then they are used only as the primary vital record specification. Open data sets that do not match to the OPEN VRS are matched to other vital record specifications, as normal. . If you need to use the OPEN job names as non-special values, you have to use generic job names in the vital record specification, such as OPEN*.

DFSMSrmm implements the OPEN vital record specifications as follows:

Note: DFSMSrmm does just only one of the following, but in this sequence.

- If the DSNAME mask is 'OPEN', the data set name mask of '**' is used. The JOBNAME mask is used as it is.
- If the JOBNAME mask is 'OPEN', the job name mask of '*' is used. Also, no JOBNAME mask is used in case any of the data sets do not have a job name.

Thus, it is possible to define multiple vital record specifications that result in the same masks being used, or that may not be used as expected. For example,

- DSNAME('OPEN') without job name and DSNAME('**') JOBNAME(OPEN) are equivalent, but the latter is used by DFSMSrmm. EDG2219I issued for the former because of key sequence.
- DSNAME('OPEN') JOBNAME(*) and DSNAME('**') JOBNAME(OPEN) are equivalent, but the latter is used by DFSMSrmm filter matching processing. The former is ignored, because of key sequence. Matching data sets show that OPEN vital record specification have been used.
- DSNAME('OPEN') JOBNAME(OPEN) results in a mask of DSNAME('**') JOBNAME(OPEN) that only matches cases where the job name actually is 'OPEN'.

VRSEL processing issues message EDG2219I if any conflicting definitions are created.

Name filtering with data set name masks

When DFSMSrmm is deciding the vital record specification that matches to a data set, it does so by attempting to match to the most specific mask that applies. For example, 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 uses jobname to apply policies

Table 7 on page 62 shows how DFSMSrmm applies policies based on the jobname you specify and the DFSMSrmm parmlib EDGRMMxx member OPTION VRSJOBNAME operand value you use.

Table 7. 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 8 on page 62shows the order that DFSMSrmm uses to match job names defined to DFSMSrmm to job name masks in vital record specifications.

Table 8. 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.

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 and associated management class names and vital record specification management values defined to DFSMSrmm. During vital record processing as described in *z/OS DFSMSrmm Implementation and Customization Guide*, DFSMSrmm matches data sets (not excluded from VRSEL processing) 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 VRSJOBNAME operand.

Table 9 on page 64 shows the order that DFSMSrmm uses to match a data set that are defined to DFSMSrmm to a vital record specification. If DFSMSrmm matches a data set to vital record specification based on data set name, then this is considered as the primary vital record specification. DFSMSrmm then tries to match a management class or a management value vital record specification to the data set and, if it is different from the primary vital record specification, this one becomes the secondary vital record specification. If the two vital record specifications found are the same, then they are used only as the primary vital record specification. If DFSMSrmm does not match a data set to a vital record specification based on data set name, then DFSMSrmm tries to match a management class or management value vital record specification and takes this as the primary vital record specification.

Table 9. How DFSMSrmm matches data set names to data set masks

Matching process	Examples
If a data set is on a currently opened volume, DFSMSrmm tries to select the vital record specification with the reserved job name mask or data set name mask 'OPEN' in this order:	
1. Vital record specification with job name mask 'OPEN' and data set name mask is matching to the data set name.	`RMM ADDVRS DSNAME('BACKUP.**') JOBNAME(OPEN)`
2. Vital record specification with job name mask 'OPEN' and data set name mask is matching to the management class name.	`RMM ADDVRS DSNAME('MC*') JOBNAME(OPEN)`
3. Vital record specification with job name mask 'OPEN' and data set name mask is matching to the vital record specification management value.	`RMM ADDVRS DSNAME('D99000') JOBNAME(OPEN)`

4. Vital record specification with data set name mask 'OPEN' and job name mask matching to the job name or vital record specification with job name mask 'OPEN' and data set name mask is '**'.	`RMM ADDVRS DSNAME('OPEN') JOBNAME(JOB1)`
	`RMM ADDVRS DSNAME('OPEN')`
	`RMM ADDVRS DSNAME('**') JOBNAME(OPEN)`

If a data set was created with normal disposition of DELETE, DFSMSrmm tries to select the vital record specification with the reserved job name mask or data set name mask 'DELETED' in this order:	
1. Vital record specification with job name mask 'DELETED' and data set name mask is matching to the data set name.	`RMM ADDVRS DSNAME('BACKUP.**') JOBNAME(DELETED)`
2. Vital record specification with job name mask 'DELETED' and data set name mask is matching to the management class name.	`RMM ADDVRS DSNAME('MC*') JOBNAME(DELETED)`
3. Vital record specification with job name mask 'DELETED' and data set name mask is matching to the vital record specification management value.	`RMM ADDVRS DSNAME('D99000') JOBNAME(DELETED)`

4. Vital record specification with data set name mask 'DELETED' and job name mask matching to the job name or vital record specification with job name mask 'DELETED' and data set name mask is '**'.	`RMM ADDVRS DSNAME('DELETED') JOBNAME(JOB1)`
	`RMM ADDVRS DSNAME('DELETED')`
	`RMM ADDVRS DSNAME('**') JOBNAME(DELETED)`

If a data set is closed as a result of an abnormal end, DFSMSrmm tries to select the vital record specification with the reserved job name mask or data set name mask 'ABEND' in this order:	
1. Vital record specification with job name mask 'ABEND' and data set name mask is matching to the data set name.	`RMM ADDVRS DSNAME('BACKUP.**') JOBNAME(ABEND)`
2. Vital record specification with job name mask 'ABEND' and data set name mask is matching to the management class name.	`RMM ADDVRS DSNAME('MC*') JOBNAME(ABEND)`
3. Vital record specification with job name mask 'ABEND' and data set name mask is matching to the vital record specification management value.	`RMM ADDVRS DSNAME('D99000') JOBNAME(ABEND)`

4. Vital record specification with data set name mask 'ABEND' and job name mask matching to the job name or vital record specification with job name mask 'ABEND' and data set name mask is '**'.	`RMM ADDVRS DSNAME('ABEND') JOBNAME(JOB1)`
	`RMM ADDVRS DSNAME('ABEND')`
	`RMM ADDVRS DSNAME('**') JOBNAME(ABEND)`

Table 9. How DFSMSrmm matches data set names to data set masks (continued)	
Matching process	**Examples**
If no vital record specification is selected so far, DFSMSrmm tries to select the vital record specification in this order:	
1. Vital record specification with data set name mask and job name mask is matching to the data set name and job name.	`RMM ADDVRS DSNAME('BACKUP.**') JOBNAME(JOB1)` `RMM ADDVRS DSNAME('BACK*.**')`
2. Vital record specification with data set name mask and job name mask is matching to the management class name and job name.	`RMM ADDVRS DSNAME('MC*') JOBNAME(JOB1)` `RMM ADDVRS DSNAME('MC*')`
3. Vital record specification with data set name mask and job name mask is matching to the vital record specification management value and job name.	`RMM ADDVRS DSNAME('D99000') JOBNAME(JOB1)` `RMM ADDVRS DSNAME('D99000')`
4. Vital record specification with data set name mask of '**' and job name mask is matching to the job name.	`RMM ADDVRS DSNAME('**') JOBNAME(JOB1)` `RMM ADDVRS DSNAME('**')`

Table 9 on page 64 shows the order that DFSMSrmm uses to match data sets that are defined to DFSMSrmm to your vital record specifications. If DFSMSrmm matches a data set to vital record specification based on data set name, DFSMSrmm then tries to first match the management class and then vital record specification management value for the data set to a vital record specification for management classes and management values. If a match is found on both data set name and on either management class or management value, and the vital record specifications are different, the data set is managed by both a primary and a secondary vital record specification. If they are the same, this vital record specification is taken as the primary one.

DFSMSrmm matches to a secondary vital record specification based on management class or management value only when the data set name mask is more specific than '**'. DFSMSrmm matches a data set to the most specific data set mask when multiple matches occur.

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.

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.

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 on page 66 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.

You can use the GDG parmlib option to specify:

- Whether GDGs are processed based on generation number or based on the creation order.
- How duplicate generations are handled. You can count duplicates, keep duplicates but not count them, bump duplicates from the current subchain, or drop duplicates from VRS retention

Detection of duplicate generation data sets

In the example in Figure 19 on page 66, there are five generations of a GDG, named A.B, with some using the same generation number. In the example, you can see the creation sequence from oldest to newest, and also the order in which VRSEL processing processes them from newest to oldest. DFSMSrmm first matches each data set to a VRS, then determines the subchain, if any, that retains the data set.

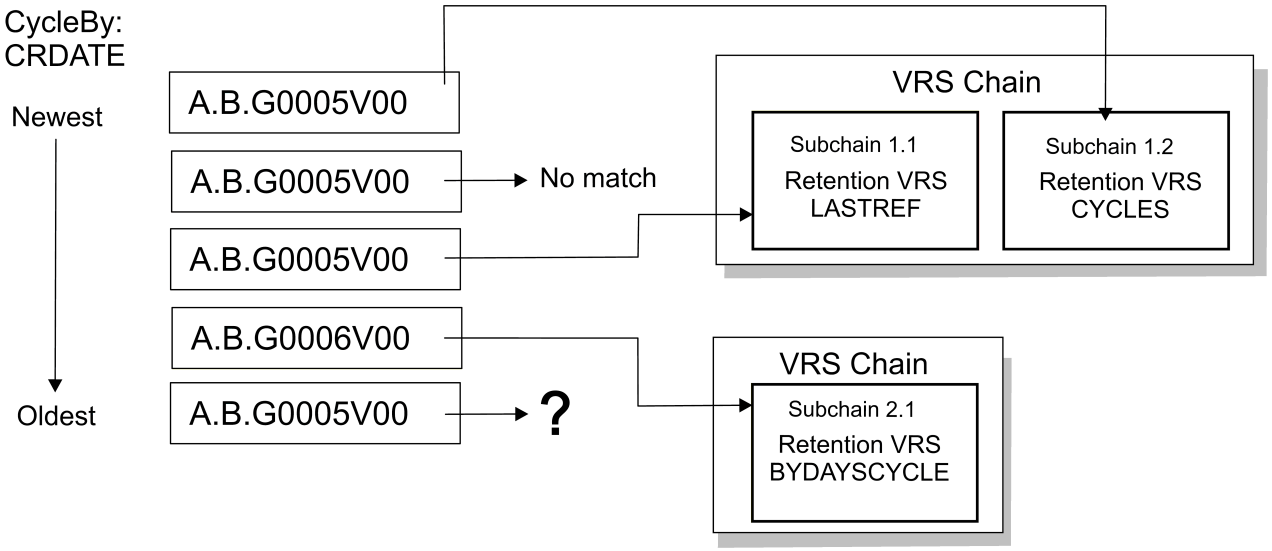


Figure 19. Duplicate GDG example

Will the oldest generation be detected as a duplicate?

For processing sequence 5, you can see that it is important which subchain the data set is retained by:

- When retained by the 1.1 subchain, which retains with LASTREFDAYS, the generation is not a duplicate because DFSMSrmm detects duplicates only for a cycle based retention.

- When retained by the 1.2 subchain, which retains with CYCLES, a newer generation using the same generation number is already retained and so the older generation is detected as a duplicate. How it is processed as a duplicate depends on the DUPLICATE operand of the GDG parmlib option, as shown in Table 10 on page 67.

Table 10. Subchain used for different GDG DUPLICATE options

Process Sequence	Data Set Name	Subchain used for different GDG DUPLICATE options			
		BUMP	DROP	KEEP	COUNT
1	A.B.G0005V00	1.2	1.2	1.2	1.2
2	A.B.G0005V00	No match	No match	No match	No match
3	A.B.G0005V00	1.1	1.1	1.1	1.1
4	A.B.G0006V00	2.1	2.1	2.1	2.1
5	A.B.G0005V00 LASTREF=true	1.1	1.1	1.1	1.1
5	A.B.G0005V00 LASTREF=false	None	None	1.2	1.2

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 20 on page 67 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 20. Specifying data set name masks for pseudo-GDGs

The restrictions for using masking characters are:

- You cannot specify ** before a qualifier with a single * character. For example, you cannot use this example.

  ```
  RMM ADDVRS DSNAME('JAMES.**.*.DIV¬')
  ```

- You cannot include a ¬ character between two * characters in each data set qualifier. For example, you cannot use this 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 this example.

  ```
  RMM ADDVRS DSNAME('JAMES.**.DIV¬.SOCCER.**')
  ```

During DFSMSrmm inventory management, DFSMSrmm recognizes the ¬ masking character, and applies the appropriate retention and movement policy that is defined for a pseudo-GDG. DFSMSrmm validates the mask that 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 21 on page 68, the vital record specification defines a policy for a user-defined set of data sets that are sequenced like a GDG.

```
RMM ADDVRS DSNAME('PRITCHAR.X¬¬¬¬Y¬¬') -
    NOGDG CYCLES COUNT(3)
```

Figure 21. 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 might use the data set name mask that is shown in Figure 22 on page 68.

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

Figure 22. 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 23 on page 68 shows an example of defining two vital record specifications for the same data set name with the CYCLES operand. When you use the CYCLES operand, the retention of duplicate GDGs is controlled by the setting for the DUPLICATE operand of the OPTION GDG in parmlib.

```
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 23. Specifying NOGDG and CYCLES in a vital record specification

Figure 24 on page 68 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 GDGs are retained regardless of the setting of the DUPLICATE operand of the OPTION GDG in parmlib.

```
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 24. 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 236 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.
4. Use the CHANGEVRS subcommand to update details of a data set mask or a NAME vital record specification.

See "LISTVRS: Displaying information about a vital record specification" on page 349, "DELETEVRS: Deleting vital record specifications" on page 318, and "ADDVRS: Adding a vital record specification" on page 236 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 318 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 5. Using vital record specifications to retain and move volumes

Use vital record specifications to define retention and movement policies for volumes managed by the VRSEL retention method and the data sets on those volumes that are not excluded from VRSEL. See Chapter 4, "Defining vital record specifications," on page 53 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" on page 73. See "Defining retention policies for data sets and volumes" on page 77 and "Defining movement policies for volumes" on page 88 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 97 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 337 to request retention information.

Types of retention

This topic 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 either by generation and version numbers or by creation order, based on the CYCLEBY operand of the GDG parmlib option. You can also use the DUPLICATE operand of the GDG parmlib option to specify how VRSEL processing handles duplicate generations. For non-GDG data sets, DFSMSrmm considers each occurrence of a data set to be a cycle.

Example

To retain five versions of the GDG, NBHART.DATA, issue the command:

```
RMM ADDVRS DSNAME('NBHART.DATA') GDG CYCLES COUNT(5)
```

BYDAYSCYCLE

Define a vital record specification with the minimum number of cycles or copies that should be kept for a matching data set. For BYDAYSCYCLE processing, DFSMSrmm sorts GDGs either by generation and version numbers or by creation order, based on the CYCLEBY operand of the GDG parmlib option. You can

© Copyright IBM Corp. 1992, 2022

also use the DUPLICATE operand of the GDG parmlib option to specify how VRSEL processing handles duplicate generations. 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:

```
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.

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 multivolume 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. If there is a secondary vital record specification, UNTILEXPIRED processing looks at the retention date of the secondary vital record specification instead of the volume expiration date.

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. However, the volume is never released prior to its expiration date, unless RELEASE(EXPIRYDATEIGNORE) is specified.

Example

To retain all data sets until the volume expiration date has been reached, issue the command:

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

By data set status (open, deleted, or abended)

Define a vital record specification to set a policy for data sets that are:

- left open by a system failure
- created with a normal disposition of DELETE, 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 for open data sets

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 for deleted data sets

To retain data sets created with a normal disposition of DELETE, issue the command:

```
RMM ADDVRS DSNAME('DELETED') LASTREFERENCEDAYS COUNT(2)
```

Example for abended data sets

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)
```

Example for data set name

To have a special policy based on a data set name, issue the command:

```
RMM ADDVRS DSNAME('**.BKUP.**') DAYS COUNT(2) JOBNAME(OPEN)
```

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 2011/361 is reached or the data set is uncataloged, issue the command:

```
RMM ADDVRS DSNAME('MIKES.DONT.DELETE.UNTIL.BTHDAY') COUNT(99999) -
    DELETEDATE(2011/361) 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 managed by the VRSEL retention method, and data set on these volumes that are not excluded from VRSEL processing. 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 these:

- 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 defaults table, specified in the EDGDEFxx parmlib member.
- The DFSMSrmm EDG_EXIT100 installation exit. Use DFSMSrmm installation exit EDG_EXIT100 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 z/OS 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 defaults table specified in the EDGDEFxx parmlib member, or the DFSMSrmm-supplied EDG_EXIT100 installation exit as described in the *z/OS DFSMSrmm Implementation and Customization Guide* to assign management values to data sets.

Figure 25 on page 78 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 25. Retaining a data set using management class

Figure 26 on page 78 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 26. 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

Normally, when a data set matches multiple retention policies, the most specific policy is used. However, when both a primary and a secondary vital record specification match to a data set, DFSMSrmm combines these retention policies. For example, this happens 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 also matches. If the '*.**' vital record specification contains UNTILEXPIRED, the retention of the data set is bounded by the management class name vital record specification. Otherwise, the data set is kept as long as either the primary or the secondary VRS is vital.

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 true for the data set to be retained. In Figure

27 on page 79, the data set is retained until the date 2011/361 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(2011/361) LOCATION(REMOTE) WHILECATALOG
```

Figure 27. 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 28 on page 79 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 28. Using UNTILEXPIRED and WHILECATALOG on a single vital record specification

UNTILEXPIRED can be used with all retention types.

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.

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.

DFSMSrmm handling of the vital record specification release options enables them to be applied to a volume regardless of whether any data set on the volume is retained by a vital record specification. Release options are applied for any data set that matches to a vital record specification. This means that you can return a data set to scratch on the same day that it is created.

When there are multiple data sets on a volume, the results for release option processing are such that:

- If any data set on a volume is or has been retained by a vital record specification, the release options for the volume are set only from data sets that are retained by that vital record specification.
- If no data sets on a volume are vital record specification retained and none of them have yet been retained by a vital record specification, the release options are taken from any data sets that match to a vital record specification. Both primary and secondary vital record specification matches are considered.

You can specify that a vital record specification is not to retain a data set by use of COUNT(0). More additional information on using COUNT(0), see “Separating the data set name filter from the policy itself” on page 102.

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 that 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. Likewise, if a release option is specified, DFSMSrmm sets the volume release options. If on a subsequent run of inventory management, fewer or different data set match and still no data set is yet retained as a vital record, this process is repeated until the volume is released and the release options from the last run of inventory management are implemented.

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. You can have two vital record specifications match to a volume.

If you define a vital record specification with the '**' data set name mask, 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, 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 4, “Defining vital record specifications,” on page 53 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.

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 primary vital record specification when it is vital, and the secondary vital record specification when the primary vital record specification no longer retains the data set. 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.

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 make sure that the primary vital record specification does not retain the data set past the retention of the secondary vital record specification. While the primary vital record specification controls the data set, the displayed retention date is the earliest of the two dates.

Retention date format

DFSMSrmm displays retention date in date format as shown in Table 11 on page 81 or with DFSMSrmm special date formats.

Table 11. Retention 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 parmlib option CATRETPD(12) 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 parmlib option CATRETPD(12) 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 PM. DFSMSrmm retains

the data set rather than releasing it when DFSMSrmm inventory management is run (12:00 PM through 2:00 PM).

```
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) operand to ensure that the retention specified in the VRS overrides the volume expiration date. Table 12 on page 83 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: 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, DFSMSrmm uses the earliest of the two retention dates for determining how long the primary vital record specification retains the data set.

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 this order:

1. PERMANENT
2. CYCL/*ccccc*
3. WHILECATLG

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

Table 12 on page 83 and Table 13 on page 84 describe how DFSMSrmm calculates the retention date.

Table 12. 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" on page 84	CYCL/ccccc
DAYS	COUNT plus the create date"1" on page 84	Date format specified by your installation
EXTRADAYS (XD)	COUNT plus the date"1" on page 84"4" on page 84	Date format specified by your installation
LASTREF	COUNT plus the last reference date"1" on page 84	Date format specified by your installation
CYCLES + WC	Special catalog date format"2" on page 84	WHILECATLG
DAYS + WC	COUNT plus the create date"1" on page 84	Date format specified by your installation
LASTREF + WC	COUNT plus the last reference date"1" on page 84	Date format specified by your installation
CYCLES + UEX	Volume expiration date"1" on page 84	Date format specified by your installation
DAYS + UEX	Lower of volume expiration date and COUNT plus create date"1" on page 84	Date format specified by your installation
LASTREF + UEX	Lower of volume expiration date and COUNT plus date last referenced"1" on page 84	Date format specified by your installation
CYCLES + WC + UEX	Volume expiration date"1" on page 84	Date format specified by your installation
DAYS + WC + UEX	Lower of volume expiration date and COUNT plus create date"1" on page 84	Date format specified by your installation
LASTREF + WC + UEX	Lower of volume expiration date and COUNT plus date last referenced"1" on page 84	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 for the retention of the DSN subchain."1" on page 84,"3" on page 84	As determined by the vital record specification retention options.

Table 12. 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. 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.
3. The retention date format can be a date or any of the special date formats.
4. DFSMSrmm uses the date when the subchain started to retain the data set as the retention date.
5. In the case there is a data set(s) on the volume that matches a vital specification along with one or more data sets on the volume that are not matched to vital record specifications, the release options of the matched data set will be honored. This could lead to premature expiration of the non matched data set. The recommendation is that if Vital record specification(VRS) expiration method is being used; all data sets should have a VRS policy defined. Normally a default VRS policy for dsn ** will prevent this issue and DFSMSrmm will assign the highest retention date from all data sets managed by VRS for the volume or volume set.

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 13 on page 84 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 13. 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" on page 86	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

Table 13. DFSMSrmm retention date calculation by COUNT(99999) (continued)

If you have retention type:	RMM calculates retention date as:	And displays retention date as:
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" on page 86	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
DAYS + WC	vital record specification delete date OR catalog date format"2" on page 86	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
LASTREF + WC	vital record specification delete date OR catalog date format"2" on page 86	Date format specified by your installation or WHILECATLG if WHILECATALOG is specified.
CYCLES + UEX	Volume expiration date"3" on page 86	Date format specified by your installation
DAYS + UEX	Volume expiration date"3" on page 86	Date format specified by your installation
LASTREF + UEX	Volume expiration date"3" on page 86	Date format specified by your installation
CYCLES + WC + UEX	Volume expiration date"3" on page 86	Date format specified by your installation
DAYS + WC + UEX	Volume expiration date"3" on page 86	Date format specified by your installation
LASTREF + WC + UEX	Volume expiration date"3" on page 86	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. "2" on page 86, "4" on page 86	As determined by the vital record specification retention options.

Table 13. 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. DFSMSrmm calculates retention date based on both vital record specifications and information in this table.
5. In the case there is a data set(s) on the volume that matches a vital specification along with one or more data sets on the volume that are not matched to vital record specifications, the release options of the matched data set will be honored. This could lead to premature expiration of the non matched data set. The recommendation is that if Vital record specification(VRS) expiration method is being used; all data sets should have a VRS policy defined. Normally a default VRS policy for dsn ** will prevent this issue and DFSMSrmm will assign the highest retention date from all data sets managed by VRS for the volume or volume set.

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 87 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 on page 87. 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 on page 87.

```
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.

Converting from expiration date or cycle control to using catalog control

Recommendation: With expiration date or cycle control, you might be retaining tape volumes that should be scratched when their GDG limit is decreased (because the expiration date or cycle number has not been reached). Consider moving to a catalog-control-based system to manage your tape volumes based on the GDG base defined in the catalog.

1. Use the REPORT file created by VRSEL to determine whether a VRS retains GDGs or regular data sets.

   For a VRS that retains non-GDG (regular) data sets, care taken when converting from COUNT(*n*) to WHILECATALOG, because for non-GDG data sets there can be can only one data set of that name cataloged, so if a VRS that specifies COUNT(3) is changed to COUNT(3) WHILECATALOG, only one data set will be retained.

   For a VRS that retains only GDG data sets, you can add the WHILECATALOG option to the VRS and then run VRSEL with the VERIFY option. Analyze the ACTIVITY file with EDGJACTP to verify that the results are satisfactory. If so, you can then alter the COUNT value to COUNT(99999) to cause DFSMSrmm to rely on the catalog status.
2. Modify the vital record specifications shown in Figure 31 on page 87.

```
RMM ADDVRS DSNAME('SAMPLE.GDG') GDG CYCLES COUNT(1) LOCATION(ATL1) -
    NEXTVRS(N1) WHILECATALOG
RMM ADDVRS NAME(N1) COUNT(99999) CYCLES WHILECATALOG LOCATION(REMOTE)
```

Figure 31. Managing volumes using catalog control

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 91for more information.

The EDGRRPTE REXX report, REPORT17, is a helpful tool to aid with stacked volume management. See *z/OS DFSMSrmm Reporting* for additional information on REPORT17.

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 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 269 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 can specify a distributed library name only if the library is an IBM Virtualization Engine.

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 these 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 32 on page 89 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 32. Moving volumes out of a system-managed library

The vital record specifications shown in Figure 32 on page 89 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 32 on page 89, 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 32 on page 89, 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 32 on page 89 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 33 on page 89 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 33. Moving a volume in and out of the system-managed library

In Figure 33 on page 89 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 34 on page 90 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 34. Moving data out of the system-managed library at fixed intervals

In Figure 34 on page 90, 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 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.

Volume movement can also be controlled by location. With the DFSMSrmm parmlib LOCDEF AUTOMOVE operand, you can define locations that are not applicable for automated movement.

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 35 on page 91.

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

Figure 35. 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 35 on page 91 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 36. Sending a volume to another system

The MANUALMOVE operand shown in Figure 36 on page 91 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 36 on page 91 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 37 on page 91.

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

Figure 37. 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 95 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 8, “Requesting information about your resources,” on page 129 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 269 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 z/OS 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. This default processing can

be customized by using DFSMSrmm parmlib commands to define how volumes should be partitioned. You can use either PRTITION or REJECT ANYUSE commands to define partitioning. REJECT is superseded by the PRTITION command. Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for more information about partitioning.

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 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 38 on page 93 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 38. 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 CHANGEVOLUMEvolserRACK(rack_number)
  ```

  where:

  pool ID is one-to-five characters that is followed by an * and 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 39 on page 95 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 39. Moving a volume to a loan location

Figure 40 on page 95 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 40. 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 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 41 on page 95 to clear the loan location value.

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

Figure 41. Returning a volume from a loan location

See “CHANGEVOLUME: Changing volume information” on page 269 and “SEARCHVOLUME: Creating a list of volumes” on page 380 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 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 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 42 on page 96:

```
 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 42. DFSMSrmm Volume Action Status panel

3. Type C next to the type of move you want to confirm as shown in Figure 42 on page 96. 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 43 on page 97:

```
 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 43. 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 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 44 on page 98 shows how to confirm all moves from location REMOTE to location SHELF that are ready to scratch.

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

Figure 44. 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 45 on page 98.

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

Figure 45. 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 380 for more information on using SEARCHVOLUME with the CLIST operand. See “Creating CLISTs of executable subcommands” on page 146 for more information on creating lists of executable subcommands.

Chapter 6. Using vital record specifications to define 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 topic 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 and processing options by using the parmlib member EDGRMMxx OPTION command operands RETPD, CATRETPD, MAXRETPD, and EXPDTDROP.
2. Set vital record processing options by specifying the parmlib member EDGRMMxx OPTION command operands GDG, MOVEBY, RETAINBY, VRSCHANGE, VRSDROP, VRSJOBNAME, VRSMIN, and VRSRETAIN.
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 these references before defining policies to DFSMSrmm.

- Chapter 4, “Defining vital record specifications,” on page 53 and Chapter 5, “Using vital record specifications to retain and move volumes,” on page 73 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 4, “Defining vital record specifications,” on page 53 describes how to define vital record specifications by using the RMM ISPF dialog.
- Chapter 10, “Using RMM TSO subcommands,” on page 187 describes these RMM TSO subcommands: ADDVRS, CHANGEVRS, DELETEVRS, LISTVRS, and SEARCHVRS.
- *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 EXPDTDROP, VRSCHANGE, VRSDROP, VRSJOBNAME, VRSMIN, and VRSRETAIN are used to control how DFSMSrmm processes retention and movement policies. The GDG operand is used to control how generation data groups are handled for cycle retention by VRSEL processing. 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 79 for information about using the DFSMSrmm EDGRMMxx parmlib OPTION RETAINBY and MOVEBY operands to define how volumes are retained or moved.

© Copyright IBM Corp. 1992, 2022

- Describes DFSMSrmm inventory management vital record processing that performed for retention and movement policies to take effect.
- Provides vital record specification examples for retaining and moving DFSMShsm data sets.
- Provides information about using the DFSMSrmm EDG_EXIT100 exit to assign vital record management values to data sets.

What vital records selection processing you can specify

When specifying your vital records selection processing, you can:

- Continue to use any vital record specifications you have defined previously and to expect that there is no change to how those policies are applied.
- Define vital record specifications with data set and volume names that are based on the matching order described in Table 9 on page 64.
- Define vital record specifications including retention information on a name vital record specification chained using the NEXTVRS or ANDVRS subcommand operands, so that each vital record specification can have a different retention type. Such chains describe how to move data through several locations using one or more retention types.

  DFSMSrmm applies policies one vital record specification subchain at a time when the retention criteria is true for a data set.
- Define vital record specification for VRS management values and SMS management classes and use SMS ACS routines and the DFSMSrmm EDG_EXIT100 installation exit to assign these values so retention and movement policies can be applied to data sets and volumes.
- UNTILEXPIRED allows primary vital record specifications to manage location, without retaining data sets longer that they would be retained by the secondary vital record specification.
- Use the COUNT(0) operand to specify that a vital record specification is not to retain a data set.
- Apply vital record specification release options to a volume regardless of whether any data set on the volume is retained by a vital record specification. Release options are applied for any data set that matches to a vital record specification. This means that you can return a data set to scratch on the same day that it is created.
- Use either the special ABEND, DELETED, or OPEN data set name mask with a JOBNAME to select special retention, or use the special ABEND, DELETED, or OPEN job name with any data set name mask. For example, see Figure 46 on page 100:

```
...
RMM AS DSNAME('HSM.**') DAYS COUNT(99999) JOBNAME(OPEN)
RMM AS DSNAME('HSM.**') DAYS COUNT(99999) JOBNAME(DELETED)
RMM AS DSNAME('HSM.**') DAYS COUNT(99999) JOBNAME(ABEND)
RMM AS DSNAME('OPEN') DAYS COUNT(99999) JOBNAME(HSM)
RMM AS DSNAME('MCLASS01') COUNT(0) JOBNAME(ABEND) NEXTVRS(ABEND2)
 ...
```

Figure 46. Using special ABEND, DELETED, or OPEN job names

You can combine the use of ABEND and OPEN vital record specifications with COUNT(0).

DFSMShsm considerations: Special considerations apply to DFSMShsm-owned volumes in the event of an ABEND or OPEN. If you want to hold these volumes after an ABEND or OPEN, you must define a specific data set name VRS. This is particularly needed if you also have defined a generic VRS that sends abended volumes to scratch immediately (for example, by COUNT(0)). See Figure 46 on page 100 for a sample VRS to permanently hold abended DFSMShsm-owned volumes. Note that the data set name mask must be adjusted to your installation. These considerations also apply to other external data management applications. For more information, see the topic "Running DFSMSrmm with DFSMShsm" in *z/OS DFSMSrmm Implementation and Customization Guide*.

- Manage your vital record specifications based on information provided by DFSMSrmm. DFSMSrmm identifies the vital record specification policy chains that are not being used and provides a last

reference date and time for each vital record specification you define. Every time that you run VRSEL processing and DFSMSrmm attempts to retain a data set or volume to a vital record specification subchain, the last reference date and time of the vital record specification records is updated to the VRSEL run date and time. Also, at the end of VRSEL processing, the control data set records are updated if referenced during that run, and the last reference date and time is updated. At the end of the run, the unused vital record specification chains are listed in the REPORT file and counts of the unused vital record specification records are printed in the MESSAGE file.

- 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.
- 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. When you specify COUNT(0), it does not matter what location or retention criteria you specify for the vital record specification, because these are never considered by DFSMSrmm.
- Manage data set retention and movement separately. Use a vital record specification based on VRS management value or SMS management class name to define retention criteria; use a vital record specification based on data set name mask for defining movement.
- 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 114 for information about release actions.
- Request that volumes are returned to scratch status in one expiration processing run when the only release action is return to scratch.
- 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.
- Define parmlib options to set your retention expectations and have DFSMSrmm generate alerts when the expectations are not met.

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 vital record processing and expiration processing. We recommend running inventory management vital record processing and expiration processing in a single job step.

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. A subchain includes all the vital record specifications chained from the start of the subchain until the next subchain starts. 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 volume vital record specifications with one or more name vital record specifications. The entire vital record specification chain is considered the policy for the data set.

For example, you can create a vital record specification chain for 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.

You can specify vital record specifications, as follows:

- Any volume vital record specification can only contain movement information.
- 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.

Separating the data set name filter from the policy itself

While it is possible to create data set name filter VRSs that also contain retention and movement information, doing so will eventually result in many duplicate policy definitions. If you ever need to change any of your policies, you will need to locate and update many different copies of the same policy. Also, if these policies are separately defined, you run the risk of inconsistent policies being defined for data sets that should all be using the same policy.

You can avoid these problems by separating the data set name filter from the policy itself, which enables clear and well-defined service levels to be set up for tape management. These policy and service-level VRSs can then be easily modified as required without changing the filters that select them. You can achieve this separation by creating a data set name VRS that uses a COUNT of zero, so that the first VRS in a chain has no retention specification. The NEXTVRS in the chain and subsequent vital record specifications are then used to specify the complete retention policy. To do this when adding a new data set to VRS retention:

1. Select an existing service level represented by a VRS chain that starts with a retention NAME VRS.
2. Define the data set name VRS, specifying DSNAME, JOBNAME, and GDG or NOGDG as required, with COUNT(0) and NEXTVRS(*name*), where NEXTVRS(*name*) identifies the beginning of a subchain that specifies the desired policy.

For example, suppose you want to assign a new data set name to an off-site retention policy to retain data for 365 days and then until expired and that you have an existing policy specified by a VRS subchain named YEAROFF:

```
RMM ADDVRS NAME(YEAROFF) COUNT(365) DAYS LOCATION(OFFS) NEXTVRS(UEX)
RMM ADDVRS NAME(UEX) UNTILEXPIRED
```

To apply this policy to the new data set name, specify:

```
RMM ADDVRS DSNAME('NEW.DATASET.**') NOGDG JOBNAME(*) NEXTVRS(YEAROFF) COUNT(0)
```

Using COUNT(0) in a VRS specifies that the VRS in question cannot retain a data set, and that VRS is to continue with the next VRS in the chain, if any. In the same way, when you specify COUNT(0) as part of an ANDVRS group, the ANDVRS group cannot retain a data set. When you specify COUNT(0), any location or retention criteria you specify for the VRS is ignored by DFSMSrmm.

COUNT(0) can be specified with the ADDVRS or CHANGEVRS subcommands. You can also use the EDGPV200 panel (DFSMSrmm Add Vital Record Specification) to create a data set name filter by specifying YES in the "Add data set filter VRS" field. The added VRS will be set with COUNT=0 and chain type NEXT.

If you have existing data set name VRSs that also contain policy information and you wish to convert them to separate the data set name filter from the policy information, you can use the EDGRVCLN REXX procedure. The LIST(FILTER) parameter of EDGRVCLN will provide a list of data set VRSs that are candidates for using COUNT(0). The FIX(FILTER) parameter of EDGRVCLN will implement the changes recommended by LIST(FILTER). See "EDGRVCLN REXX procedure to clean up vital record specifications" on page 110 for additional information on using EDGRVCLN.

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 47 on page 103 shows two vital record specifications for retaining DFSMShsm data sets.

Figure 48 on page 104 and Figure 49 on page 104 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 47. 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 48. Specifying a vital record specification using the dialog part 1

Figure 49 on page 104 shows the same information as shown in the second vital record specification defined in Figure 47 on page 103.

```
 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 49. 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
```

DFSMSrmm can match two vital record specifications to a data set. If you define a vital record specification with the '**' data set name mask, 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,

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 67.

Writing your own vital record specifications

To write your own vital record specifications:

1. Review the examples in this topic.
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 non-GDG data sets

Retain the latest 3 cycles of data sets that match to the data set name mask in the home location. For non-GDG data sets, a 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('NG.**')    -              /* 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 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 GnnnnVnnn. 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.

These data sets are created in this 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 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           */
```

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 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 EDG_EXIT100 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
```

Maintaining your vital record specifications

You may need to update your vital record specifications, either to exploit a new function available in the latest release or to make policy changes based on the needs of your business. You can perform these steps to ensure that these changes go smoothly.

1. Before updating your vital record specifications, 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 vital record specifications” on page 110 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. Make sure that all systems sharing DFSMSrmm control data sets have the same parmlib options.
5. 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. The changes that would have been made on a non-VERIFY run are included in the REPORT file and written to the ACTIVITY file.

6. 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. Verify that the ACTIVITY file records reflect only the changes you have made. If no changes have been made, the ACTIVITY file should be empty. If there are entries, run EDGJACTP from SAMPLIB and look at the summary files. Look in particular at "VRSS DD Statement - summary by status". Verify the data sets that changed from V/R to not V/R retained or vice versa. The details of why it changed is shown in "VRS DD Statement - data sets with changed V/R status". The reports produced in other DD statements will also be of interest. Be sure to review each of the summary and detail files to understand the changes that would have been made.
7. Correct vital record specifications as needed to make sure that the policies you want are in place.
8. Continue running the VERIFY function and inspect the results until you get the results you expect using the new functions
9. Begin defining and updating vital record specifications to use new or different options and retention. 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. The kind of changes you make might include VRS release options, use of COUNT(0), exploit the OPTION GDG of parmlib, consolidation to generic data set name masks, or cleanup of unused VRSs or VRS chains.
10. Repeat the process from step “5” on page 109 and step “9” on page 110 until you are satisfied that the results are what you expect.
11. Run inventory management production run processing.
12. 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 vital record specifications

You can use the EDGRVCLN REXX procedure to report on and update existing vital record specifications. This can help you with implementing new function and better exploiting existing function in vital record specifications.

EDGRVCLN provides these options that you can use to identify retention information that could be changed and to implement changes:

- LIST(CYCLES) to list all vital record specifications that specify a retention type of CYCLES.
- LIST(DSNCHAIN) to list all vital record specification chains.
- LIST(FILTER) parameter to analyze the existing vital record specifications and also to see how many different types of policies you have.

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(CURRENT) parameter to identify any vital record specifications that use the location name CURRENT, which is reserved by DFSMSrmm. You can edit the commands produced by running EDGRVCLN to change the location name from CURRENT to a new location name.
- Use the LIST(ERROR) list all name vital record specifications that contain incorrect or incomplete retention information. You can use FIX(ERROR) to correct any errors that are detected.

If you do not correct 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 these parameters:

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).

Prior to being designated as a DFSMSrmm reserved location name, the location name CURRENT was free to be used a storage location name. You may have had a LOCDEF for LOCATION(CURRENT) in the past and now removed it. Any VRSs that were added while the LOCDEF was in use, redefined to ensure the correct processing for the now reserved DFSMSrmm location name. Use the LIST(CURRENT) parameter to identify any vital record specifications that use the location name CURRENT. You can edit the commands produced by running EDGRVCLN to change the location name from CURRENT to a new location name.

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. 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.

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

The procedure searches for all data set vital record specifications. The output includes all data set vital record specifications and all name vital record specifications in the chain. You can use this option to see a consolidated list of all existing VRSs presented as chains so that you can see what policies are implemented. You can also use the commands created by the LIST option as a basis for making changes to your policies. Also refer to the LIST(FILTER) option, which does further analysis of your policies.

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.

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.

LIST(FILTER)
Use this parameter to list all data set VRSs that are candidates for exploitation of the use of COUNT(0).

The procedure searches for all data set name VRSs that do not specify COUNT(0) and accumulates together those that have common delay, retention, and movement requirements, and NEXT/ANDVRS specified in the first VRS in the chain.

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

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(FILTER)
Use this parameter to implement the changes recommend by LIST(FILTER).

The procedure creates one or more retention NAME VRSs for each of the common groups of retentions found, and then uses the CHANGEVRS subcommand to set COUNT(0) and the NEXTVRS to chain to the new NAME VRS(es) created. If DELAY was in use, a retention name VRS is created for the DELAY using DAYS since creation and a NEXTVRS to the retention name VRS(es) for the retention and location pulled from the DSNAME VRS.

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. When you use the LIST(FILTER) parameter, the LIST file includes an RMM CHANGEVRS and one or more ADDVRS subcommands for each vital record specification that is a candidate for exploitation of the use of COUNT(0). 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, changes, and adds vital record specifications as part of the FIX processing. Use the JCL in Figure 50 on page 112 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,DYNAMNBR=30
//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 50. Sample JCL to run the EDGRVCLN procedure

Chapter 7. Requesting and releasing volumes

This topic 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 320 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 these:

   - An owner ID other than your own, which is the default
   - A retention period or an expiration date if you do not want to use the default retention period set up by your installation
   - A pool ID if you want the volume to be chosen from a specific pool
3. Press ENTER.

   DFSMSrmm selects a scratch volume from either a default scratch pool or from a pool you specified, and changes the status of the volume to user status.

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

To display the security classes defined for your location, type CONTROL SECURITY from the command or option line of any panel. In the TSO environment, use the LISTCONTROL SECCLS subcommand.

Releasing volumes

Related TSO subcommand: Use the ADDVOLUME subcommand to specify release actions for a volume when you initially add it to DFSMSrmm. Use the CHANGEVOLUME subcommand to specify release actions for a volume already defined to DFSMSrmm. See "ADDVOLUME: Adding volume information" on page 211 and "CHANGEVOLUME: Changing volume information" on page 269 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 236 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.

© Copyright IBM Corp. 1992, 2022

- 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.
- The volume is not held by its HOLD attribute.
- The volume is not kept by its EDM attribute.

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 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 211 and “ADDVRS: Adding a vital record specification” on page 236 for information about defining policies to control when volumes are eligible for release.
- See “DELETEVOLUME: Deleting volume information” on page 314 for information about using the DELETEVOLUME subcommand to manually release one or more volumes.
- See “Deleting information for volumes in multivolume sets” on page 34 for information about using the DFSMSrmm ISPF dialog to release multiple volumes in a multivolume set.
- See “Using tapes created by programs providing their own tape management” on page 22 for the information about EDM-managed volumes and “CHANGEVOLUME: Changing volume information” on page 269 for the information about using the CHANGEVOLUME subcommand to set the EDM volume attribute.

Setting release actions for a volume

You only need to set release actions for volumes if you want to set different actions than those determined by DFSMSrmm processing. Normally, the DFSMSrmm default release actions, and built-in and customizable release actions, are used.

To manually set or change 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 14 on page 114 describes the release actions you can specify for a volume:

Table 14. Release actions for a volume

Release action	Type of processing
Any one of these:	
• Returning volumes to scratch status	Automatic when EDGHSKP is scheduled.
• Replacing volumes	Manual

Table 14. Release actions for a volume (continued)

Release action	Type of processing
• Returning volumes to their owners	Manual
Any of these:	
• 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 either the owner user ID and node, or a valid e-mail address, is 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, by default, identifies it as needing to be replaced as long as you have not manually released the volume and are running DFSMSrmm expiration processing. You can tailor this process by setting up your own volume replacement policies in PARMLIB. You can use your volume replacement policies to specify the limits for I/O errors, for Write Mount Counts, and for Age of the volumes. During DFSMSrmm expiration processing, the volumes are checked against these policies and when one or more of the limits is of reached or exceeded, the volumes' release action is set to REPLACE. DFSMSrmm also automatically sets the release action of a volume to REPLACE when the tape drive issues an alert that the volume should be replaced. Tape librarians must manually replace these volumes and confirm to DFSMSrmm that the action has taken place.

See *z/OS DFSMSrmm Implementation and Customization Guide* for more information on setting up volume replacement policies and on the tape drive SARS MIM alerts.

You can request that DFSMSrmm create a list of volumes to be replaced, using one of these:

- 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 380 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 119 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 380 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 IBM 3592, 3590, or similar media that are pre-formatted with servo tracks. Use of a degausser renders the volume unusable.

To create a list of volumes that need to be erased or initialized, use one of these:

- 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 380 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 320, “ADDVOLUME: Adding volume information” on page 211, and “CHANGEVOLUME: Changing volume information” on page 269 for more information.

To ensure that DFSMSrmm can send the message to the owner of the volume being released, you must specify either a user ID and node, or a valid e-mail address, 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 38, “ADDOWNER: Adding owner information” on page 202, or “CHANGEOWNER: Changing owner information” on page 265 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 these:

- 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 380 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 these 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 120 for details on manually releasing volumes. You can also use the DELETEVOLUME subcommand with the RELEASE operand; see “DELETEVOLUME: Deleting volume information” on page 314 for more information.

Defining the retention method to the volume

Every volume has a retention method, either EXPDT or VRSEL. All volumes in a volume set have the same retention method. The default retention method specified in parmlib is used, unless one is assigned by the EDG_EXIT100 installation exit or by the ADDVOLUME command. The retention method can be changed with a CHANGEVOLUME command, but only on the first volume in a volume set. See Chapter 3, “Retention methods,” on page 45 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.

Use the ADDVRS subcommand or the DFSMSrmm ISPF dialog to define vital record specification. See “ADDVRS: Adding a vital record specification” on page 236 or Chapter 5, “Using vital record specifications to retain and move volumes,” on page 73 for more information.

Defining vital record specifications to retain the volume

For volumes managed by retention method VRSEL and data sets not excluded from VRSEL processing, 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 236 or Chapter 5, “Using vital record specifications to retain and move volumes,” on page 73 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 269 or “Changing volume information” on page 31 for more information on changing the expiration date.

Note: The volume expiration date is related to the data set expiration date, as described in Chapter 3, “Retention methods,” on page 45.

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 15 on page 120 describes the line operators you can use against specific entries in the list to confirm pending release actions:

Table 15. 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 multivolume 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 51 on page 120 shows how to build a list of CHANGEVOLUME subcommands to confirm all replace actions for only those volumes requiring a replace action.

```
RMM SEARCHVOLUME VOLUME(*) OWNER(*) LIMIT(*) -
    CLIST('RMM CHANGEVOLUME ',' CONFIRMRELEASE(REPLACE)') -
    RELEASEACTION(REPLACE)
```

Figure 51. Confirming replace actions for volumes

See “SEARCHVOLUME: Creating a list of volumes” on page 380 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 314 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 114 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 18 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 31 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 16 on page 121 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 16 on page 121.

Table 16. 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 multivolume chain
O	Display owner information
R	Release volumes based on specified release actions

Table 16. DFSMSrmm release volumes line operators (continued)

Line Operator	Description
V	Display full volume details

Copying and moving tape data

You may have a need to copy or move tape data from one tape to another, for reasons such as replacing error prone media or moving to new technology. You can do this by using a tape copy utility that uses DFSMSrmm services to correctly copy and re-stack tape data sets while preserving the data set attribute settings and, optionally, set retention periods for the source data set, copy data set, or both. A tape copy application can notify DFSMSrmm that a newly created tape data set is a copy of an existing data set and that the target data set is actually the source data set and, optionally, that the source data set is no longer required. That is, the data set has effectively been moved.

A tape copy application might be a product such as IBM Tivoli Tape Optimizer (ITTO) or some JCL you create to copy a tape data set. If you use your own JCL and a utility like IEBGENER you can use the RMM CHANGEDATASET subcommand with the COPYFROM operand to complete the copying of the source data set attributes. However, a tape copy application can select the method used to communicate with DFSMSrmm.

DFSMSrmm provides two methods that support copying and moving tape data:

- Once the target data set has been created, the tape copy application can use the CHANGEDATASET COPYFROM subcommand to copy all applicable attributes from the source data set to the target data set. DFSMSrmm determines which attributes are to be copied. Retention of the source data set can be specifically set. Retention of the target volumes and data sets can be selected at the volume set level, and even switched between VRSEL and EXPDT retention methods.
- During the creation of the target data set, tape copy application programs can use the EDG_EXIT100 installation exit to copy all applicable attributes from the source data set to the target data set. Retention of the target volumes and data sets can be selected at the volume set level, and even switched between VRSEL and EXPDT retention methods.

Using the DFSMSrmm COPYFROM support you can ensure that all applicable data set attributes are copied and avoid using multiple RMM subcommands to modify only the attributes supported via the subcommands.

DFSMSrmm does a minimum of validation to ensure that the source and target data sets are the same. The validation differs depending on whether the copy is done with the CHANGEDATASET COPYFROM subcommand or the EDG_EXIT100 installation exit.

Using the CHANGEDATASET COPYFROM subcommand

For a target data set that has already been created, a tape copy application can use the CHANGEDATASET COPYFROM subcommand to copy all applicable attributes from the source data set to the target data set. DFSMSrmm determines which attributes are to be copied. Retention of the source data set can be specifically set. Retention of the target volumes and data sets can be selected at the volume set level, and even switched between VRSEL and EXPDT retention methods.

Using the CHANGEDATASET COPYFROM subcommand, you can ensure that all applicable data set attributes are copied and avoid using multiple RMM subcommands to modify only some of the attributes

The CHANGEDATASET COPYFROM subcommand identifies a single volume data set or any part of a multivolume data set. Validation is done to ensure that the source and target data sets have the same recording format and record length. You use the CHANGEDATASET subcommand once for each target data set record. For multivolume data sets, this means that you must issue the subcommand once for each volume the target data set is written on.

For example, data set MIKE.EXAMPLE is the only file on volume MW0001 and is copied from a physical tape into a virtual tape system. It is a single input volume, but now multiple output volumes VT0001,

VT0002, and VT0003. When the data has been copied, you would issue the following commands to copy the input data set record attributes to each of the output data set records:

```
RMM CD 'MIKE.EXAMPLE' VOLUME(VT0001) COPYFROM(VOLUME(MW0001))
RMM CD 'MIKE.EXAMPLE' VOLUME(VT0002) COPYFROM(VOLUME(MW0001))
RMM CD 'MIKE.EXAMPLE' VOLUME(VT0003) COPYFROM(VOLUME(MW0001))
```

When you use the COPYFROM subcommand operand, the last change information of the target data set is also updated during command processing to reflect that the command was processed.

Refer to "CHANGEDATASET: Changing data set information" on page 251 for details on using the COPYFROM operand.

Using the EDG_EXIT100 installation exit

The tape copy application can use the EDGPL100 macro to invoke the installation exit EDG_EXIT100 to copy the data set attributes from the source data set to the copy data set during OPEN processing. EDG_EXIT100 can notify DFSMSrmm that the data set being created is being copied from another. During OPEN processing the exit can identify the source data set from which DFSMSrmm will obtain all existing data set attributes, which will be used for the target data set. DFSMSrmm EOV processing ensures that the attributes are copied to all target data set records when the output data set becomes a multivolume data set.

Note:

1. The tape copy application that activates an EDG_EXIT100 exit module must satisfy the requirements enforced by the CSVDYNEX macro, such as system state and key, or APF authorized, or authorized by SAF. Alternatively, the installation has control by an update to PROGxx or issuing the SETPROG system command.
2. The tape copy application must use the installation exit to ensure that the source and target data set are one and the same.

Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for information about the EDGPL100 macro and EDG_EXIT100 exit.

Copying the data set attributes

After copying the data set attributes, all data set records of the target data set make the data set appear to be the source. DFSMSrmm copies all attributes that are not related to the physical aspects of the data set, volume, and tape drive.

Note:

1. Those attributes related to retention are subject to update by the next run of inventory management. The intention is that the copied data set will be retained in the same way as was the source data set. The actual results depend on the retention methods and policies used.
2. When you use the CHANGEDATASET COPYFROM subcommand operand, the last user change date and time information of the target data set is also updated during command processing to reflect that the command was processed.
3. After the attributes are copied, there will be no trace of the copy application or the batch job used to perform the copy, because all target data set attributes reflect the creation and use of the source data set. However, the last change information reflects the time the COPYFROM was done.

Table 17 on page 124 identifies data set attributes that are not copied or are copied only in certain cases. For each attribute, it lists the CHANGEDATASET operand (if any) that can set this attribute and the extract file field and REXX variable (if any) for that attribute.

Table 17. Data set attributes that are not copied by COPYFROM

Attribute	CHANGEDATASET operand	Extract file field	REXX Variable
ABEND processing **Note:** This attribute is copied only when it is set. The source setting is merged with the target setting.	ABEND	RDABEND	EDG@ABND
Block count	BLKCOUNT	RDBLKCNT	EDG@BLKC
Block size	BLKSIZE	RDBLKSZ	EDG@BLKS
Catalog status	n/a	RDCAT	EDG@CTLG
Compression ratio	n/a	RDCOMP_RAT	EDG@CRAT
Data class name	DATACLASS	RDDCNAME	EDG@DC
Data set name	*data_set_name*	RDDSNAME	EDG@DSN
Data set sequence number	LABELNUMBER	RDLABNO	EDG@DSEQ
Data set size	n/a	RDDSSIZE, RDSIZE	EDG@DSS6
Device number	DEVNUM	RDUNITAD	EDG@DEV
End block ID	n/a	n/a	n/a
Last change date	n/a	RDLCDATE	EDG@LCDT
Last change system	n/a	RDLCSID	EDG@LCSI
Last change time	n/a	RDLCTIME	EDG@LCTM
Last change user	n/a	RDLCUID	EDG@LCID
Last device number	n/a	RDLDEVN	EDG@LDEV
Logical record length	LRECL	RDLRECL	EDG@LRCL
Owner	n/a	RDOWNDSN	EDG@OWN
Percentage of the volume	PERCENT	RDPERCENT	EDG@DPCT
Physical file sequence number	FILESEQ	RDDSNSEQ	EDG@FILE
Physical space used	n/a	RDPHYS_SIZE	EDG@PSZ6
Record format	RECFM	RDRECFM	EDG@RCFM
Start block ID	n/a	n/a	n/a
Storage class name	STORAGECLASS	RDSCNAME	EDG@SC
Storage group	n/a	RDSGNAME	EDG@SG
Tape encryption key	BESKEY	RDBESKEY	EDG@BESK
Total block count (increased size)	TOTALBLKCOUNT	RDTOTAL_BLKCNT	EDG@BLK6
Total block count	TOTALBLKCOUNT	RDTOTAL_BLKCNT_OLD	EDG@BLKT

Table 17. Data set attributes that are not copied by COPYFROM (continued)

Attribute	CHANGEDATASET operand	Extract file field	REXX Variable
Volume serial	VOLUME	RDVOLSER	EDG@VOL
VRSEL exclusion **Note:** This attribute is copied only when both the source and target volumes are managed by RM(VRSEL)	VRSELEXCLUDE	RDVEX	EDG@VEX

Retention of the target data set

In addition to specifying how the source data set is retained, you can also specify how the target data set is retained. Using the EDG_EXIT100 installation exit, you select the retention method used for the target tape volume set and (if the VRSEL retention method is used) you can also exclude individual data sets from VRSEL processing. For best results, ensure that the retention method of the target tape volume set matches that of the source.

When data set attributes are copied, all existing VRSEL related attributes are copied and, as long as the target data set is on a volume set with the VRSEL retention method, you can expect the same results as for the source data set. When both the source and target data set use the VRSEL retention method, the VRSELEXCLUDE attribute is copied.

If you plan to switch data sets from the VRSEL retention method to the EXPDT retention method, ensure that the expiration date or retention period is set appropriately for each target data set.

Retention of source data sets managed by VRSEL retention method

The tape copy application can optionally decide how to manage the retention of the source data set and volume. To manage the retention of the source data set and volume, the tape copy application must issue volume and data set subcommands to set the required retention policy. The application can decide whether just the specific data set is no longer required, or the volume or the entire volume set is no longer required. Retention of data sets is subject to retention of the volume. Only when all data sets are no longer VRS retained and the volume expiration date is reached or ignored will the volume be set pending release. The choices include the following:

Keep the current retention
Assuming the target data set matches to the same VRS as the source data set, this means that the target data set and the source data set being possibly retained by the same VRS. When the copy and the source data sets use the same data set name, consider that there are two generations/cycles of the data set and if managed by cycles retention (the VRS specifies CYCLES or BYDAYSCYCLE) the retention could be unpredictable. If the data sets are generation data sets (GDSs) the setting of the DFSMSrmm parmlib option OPTION GDG(DUPLICATE()) determines how the two data sets are managed. For other data sets, the data sets might count as two cycles, potentially causing roll-off of the oldest cycle, which may be neither the source nor the target.

Use a new retention method for the data set
If you want to retain a data set currently managed by the VRSEL retention method for a defined period, you can change to use the EXPDT retention method for the data set's volume and set a retention period:

```
RMM CV volser RETENTIONMETHOD(EXPDT) RETPD(days)
```

If the volume is part of a multivolume set, you must change the retention method for the entire multivolume set by:

1. Changing the retention method for the first volume in the multivolume set. DFSMSrmm changes the retention method for all volumes in the multivolume set.

```
RMM CV volser RETENTIONMETHOD(EXPDT)
```

2. Set the required expiration date for each volume in the multivolume set:

```
RMM CV volser RETPD(days)/EXPDT(date)
```

This can be performed for the multivolume set as follows:

```
RMM SV VOLUME(volser) CHAIN LIMIT(*) CLIST('RMM CV ','
RETPD(days)')
EXEC RMM.EXEC
```

Release the volume or volume set

When copying all the files on a volume, you can decide to manage the retention or expiration at the volume level. If you no longer require the data, you can release the volume manually with:

```
RMM DV volser RELEASE
```

To release the volume chain, you can use DFSMSrmm's SEARCH with CLIST capability:

```
RMM SV VOLUME(volser) CHAIN LIMIT(*) CLIST('RMM DV ',' RELEASE')
EXEC RMM.EXEC
```

Prevent Expiration of the volume

Set the hold attribute for the source volume to prevent it being released until the data is no longer needed:

```
RMM CV volser HOLD
```

Delete the source data set

Mark the source data set as deleted and ensure that a DELETED VRS exists to manage deleted data sets. The specified *retpd* value is used to set the data set EXPDT, and if it is greater than the volume EXPDT, DFSMSrmm will increase the volume EXPDT to match. Whether this affects how long the source data is retained depends on the matching VRS. The COUNT(*retpd*) value and the retention type on the DELETED VRS will ultimately determine how long the deleted data set is retained. For example:

```
RMM CD dsname SEQ(seq) VOLUME(volume) DELETED(YES)
RMM AS DSN('DELETED') LOCATION(CURRENT) LASTREF COUNT(retpd) RELEASE(EXPIRYDATEIGNORE)
```

Exclude the source data set from VRSEL

Exclude the source data set from VRSEL by:

```
RMM CD dsname VOLUME(volser) SEQ(seq) VRSELEXCLUDE(YES)
```

You can optionally specify the RETPD operand, which sets the data set expiration date and also updates the volume expiration date, if it is a higher value than the current volume expiration date.

You can combine this with the copying of attributes in a single command:

```
RMM CD datasetname VOLUME(tgtvol) COPYFROM(VOLUME(volser) RETPD(3) VRSELEXCLUDE)
```

Retention of source data sets managed by EXPDT retention method

The tape copy application must decide how to manage the retention of the source data set and volume. To manage the retention of the source data set and volume the tape copy application must issue volume and data set subcommands to set the required retention policy. The application can decide whether just the specific data set is no longer required, or the volume, or the entire volume set is no longer required. Retention of data sets is actually subject to retention of the volume. Only when the volume expiration date is reached will the volume be set pending release. The choices include the following:

Keep the current retention

The data sets and volumes continue to be retained as they were prior to the copy. You can later decide when you want to release the volumes. Because the EXPDT retention method does not support cycle retention, no other data set can affect the retention of the source data.

Release the volume or volume set

When copying all the files on a volume, you can decide to manage the retention and expiration at the volume level. If you no longer require the data, you can release the volume manually

```
RMM DV volser RELEASE
```

To release the volume chain, you can use DFSMSrmm's SEARCH with CLIST capability:

```
RMM SV VOLUME(volser) CHAIN LIMIT(*) CLIST('RMM DV ',' RELEASE')
EXEC RMM.EXEC.CLIST
```

Retain the volume for a defined period

If you want to retain the volume for a defined period you can update the existing volume expiration date using a retention period:

```
RMM CV volser RETPD(days)
```

You can change the retention for an entire volume set by:

```
RMM SV VOLUME(volser) CHAIN LIMIT(*) CLIST('RMM CV ',' RETPD(days)')
EXEC RMM.EXEC.CLIST
```

Set a new retention policy for the data set

Data set on volumes managed by the EXPDT method can be expired after a specified period of time by including the RETPD of the COPYFROM suboperand. The source data set records are updated to enable expiration to occur. The data set EXPDT is set to the specified date. When you specify the COPYFROM RETPD operand to set the source data set expiration date, it will also update the volume expiration date if it is a higher value than the current volume expiration date.

Chapter 8. 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" on page 129.

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 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.

Table 18 on page 129 describes what to specify on the different display panels to request information about the resources defined to DFSMSrmm.

Table 18. 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.

© Copyright IBM Corp. 1992, 2022

Table 18. Information to specify on display panels (continued)

On the display panel for:	Specify:
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 52 on page 130.

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

Volume serial . . . A00001









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

Figure 52. DFSMSrmm Volume Display panel

2. Specify the unique number or name of the resource for which you want information. This number or name already defined to DFSMSrmm.

   For example, enter a volume serial number A00001 to request information about volume A00001, as shown in Figure 52 on page 130.
3. Press ENTER.

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

Managing SEARCH with large results lists

You can break down the results of a search command into manageable quantities without the need to use the END operand on a SEARCHVOLUME or SEARCHOWNER subcommand. DFSMSrmm manages the continuation of search results, and you use the LIMIT operand to specify how many search result entries you can manage each time you continue the search. For example, to search through all volumes, issue the following command:

```
RMM SV VOLUME(*) OWNER(*) LIMIT(1000) CONTINUE
```

Process the volumes that are returned. If more records exist, repeat the following command until all results are returned:

```
RMM SV VOLUME(*) OWNER(*) LIMIT(1000) CONTINUE(continue_information)
```

The *continue_information* is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged (for example, just changing the CONTINUE value on each additional command required). When there are no more entries to return, there is no *continue_information* and EDG@CONT is null/blank and message EDG3025I is not issued.

When you issue a subcommand that you know you may wish to continue, you specify the CONTINUE operand without any value. For example:

```
RMM SV VOLUME(*) LIMIT(1000) CONTINUE
or
RMM SD DSNAME(*) OWNER(MIKE) LIMIT(50) CONTINUE()
```

When issued from REXX and you request results as REXX variables, the EDG@CONT variable contains the character string you should use to continue the subcommand. When you use the API, and data is requested back as XML or SFI, the CONT SFI or XML attribute contains the character string you should use. When CONTINUE is used and the results are returned in line mode, the character string is returned as the text of message EDG3025I.

When you use the DFSMSrmm ISPF dialog to list resources, the CONTINUE support is exploited to help avoid problems with system resources. See “Using resource lists” on page 132 for additional information.

Note: Any *continue_information* you enter is handled "as is" by the DFSMSrmm command processor. The case of the variable you specify is retained and used by DFSMSrmm (for example, CONTINUE(OWNER(ae00)) and CONTINUE(OWNER(AE00)) are different continuation points). DFSMSrmm does not fold any values in the CONTINUE operand to uppercase letters. DFSMSrmm returns *continue_information* values delimited with quotes. However, you do not always need to specify the quotes. Quotes are required only when values contain special characters.

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 327 for more information.

To display the panel shown in Figure 53 on page 132, you could enter a fast path command from the command or option line of any panel.

```
EDGPC200              DFSMSrmm System Options Display
Command ===>

                                                                      More:     +
Parmlib suffix . : EN

Operating mode . : PROTECT

Data sets:
  Control  . . . : RMMUSER.APAR.MASTER
  Journal  . . . : RMMUSER.APAR.JOURNAL
  CDS id . . . . : SJTEST                   Journal threshold . : 50 %
  Catalog SYSID  : NOTSET                   Journal transaction : NO

Retention period:                           VRSDROP count  . . : 10%
  Default  . . . : 5                        VRSDROP action . . : INFO
  Maximum  . . . : NOLIMIT                  Retain count . . . : 80%
  Catalog  . . . : 1      hours             Retain action  . . : INFO
                                            EXPDT drop count . : 10%
                                            EXPDT drop action  : INFO
Use of Management Class Attributes: NONE
Retention method : Default: EXPDT

RM(VRSEL) defaults:                         RM(EXPDT) defaults:
  Retain by . . . . : VOLUME          Retain by  . . . . . . . : VOLUME
  Move by . . . . . : VOLUME
  VRS selection . . : NEW             Last Reference . . . . . : 0
  VRS Change  . . . : INFO            Catalog Days . . . . . . : 2
  VRSmin count  . . : 1               GDG     : WHILECATALOG . : OFF
  VRSmin action . . : FAIL                      RETPD  . . . . : 5
  Job name  . . . . : 2               Non-GDG : WHILECATALOG . : OFF
  GDG cycle by  . . : GENERATION                RETPD  . . . . : 5
  GDG duplicate . . : BUMP

Miscellaneous: :                            PDA: ON
  RACF support . . . . . : NONE               Block count . . . : 255
  MAXHOLD limit  . . . . : 250                Block size  . . . : 27
  IPL check  . . . . . . : NO                 Log . . . . . . . : ON
  Uncatalog  . . . . . . : SCRATCH

Enter SCROLL commands for more information, or END to CANCEL
```

Figure 53. 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 134 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 142 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 10, “Using RMM TSO subcommands,” on page 187 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 19 on page 133 describes how to select the various DFSMSrmm ISPF dialog search panels.

Table 19. 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.
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, location 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 134 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 142 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 this example:

```
There is not enough storage available to list all the volumes.
```

To avoid problems with system resources, DFSMSrmm exploits the CONTINUE processing of the SEARCH subcommands. This is done transparently by the DFSMSrmm ISPF dialog when you specify a limit value of '*', or you specify a limit value larger than 2000. The DFSMSrmm ISPF dialog issues multiple commands and returns a resource list containing the results. If there are still problems with system resources, you can do one of the following to obtain all available records from a search:

- When you use the CLIST option you can avoid the creation of a search results list.
- Check to see if the TSO profile is set to use VARSTORAGE(HIGH).
- 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 these 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.
- If you prefer to use an RMM search panel, you can use the SAVE ON panel command to create TSO SEARCH subcommands that can be processed 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

These topics 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 these:

- 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 54 on page 135:

```
 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.
```

Figure 54. 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 55 on page 135 shows:

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

     Enter the relative sort priority ( 1 to 7 ) and direction ( A or D ):

   Table field name             Priority     Direction
   ----------------             --------     ---------
   Data set name                   1             A
   Volume serial                   2             A
   Owner                           3             A
   File sequence                   4             A
   Create date                     5             A
   Expiration date
   VRS retained




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

Figure 55. 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 20 on page 136 describes the different table field names you can sort on and identifies the lists in which they are used.

Table 20. Table field names on DFSMSrmm lists

Table field name	Sort command parameter	Description	Type of list
Action	Action	Type of release action (E-erase, I-initialize, N-notify owner, O-return to owner, S-return to scratch, R-replace)	Volume action and movement
Assigned Date	Assigned	Date the volume was assigned to a user or returned to scratch status	Volume, volume action and movement
Attributes	Attributes	Special attributes associated with the volume (NONE, RDCOMPACT)	Volume
Cataloged	Cataloged	Whether or not the data set is cataloged (Y or N)	Vital record specification
Compaction	Compaction	Type of compaction technique (NONE, IDRC)	Volume
Count	Count	Number of days or number of cycles of a data set or volume should be retained	VRS
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 or data set is to be considered for release	Volume, volume action and movement Data Set

Table 20. Table field names on DFSMSrmm lists (continued)

Table field name	Sort command parameter	Description	Type of list
Feature Code	Feature	Software product feature code	Software product
File Sequence Number	Fileseq	Relative position of the data set on the volume where it resides. A decimal number between 1 and 65535.	Data set
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
Last reference Extra Days	LREF	Number of days that the data set will be retained after the data set was last referenced. A decimal number from 0 to 93000	Data set
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
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

Table 20. Table field names on DFSMSrmm lists (continued)

Table field name	Sort command parameter	Description	Type of list
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
Retention information	Retention	Combines information on Retentionmethod, RetainBy and Set Retained values. Retentionmethod is either VRSEL or EXPDT. RetainBy is provided only for RM(EXPDT) volumes and can be VOLUME, SET, or FIRSTFILE. Set Retained is provided only for RM(VRSEL) volumes and is either Y(es) or blank.	Volume
Sequence	Sequence	Data set sequence number	Data set
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
Store number	Store number	Number of days or number of cycles of a data set or volume should be retained in the location specified in the Location field	VRS
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

Table 20. Table field names on DFSMSrmm lists (continued)

Table field name	Sort command parameter	Description	Type of list
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
VRS retained	VRS retained	Whether the data set is VRS-retained (Y or blank, if not retained)	Data set

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 (upper case letters), as shown in these syntax diagrams.

Rack and bin number lists

Figure 56 on page 140 describes the syntax to sort a list of rack or bin numbers.

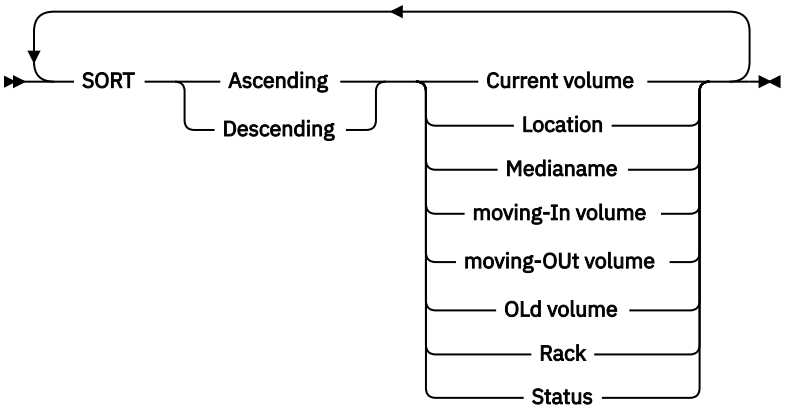


Figure 56. SORT command syntax diagram for rack or bin number lists

Volume lists

Figure 57 on page 140 describes the syntax to sort a list of volumes.

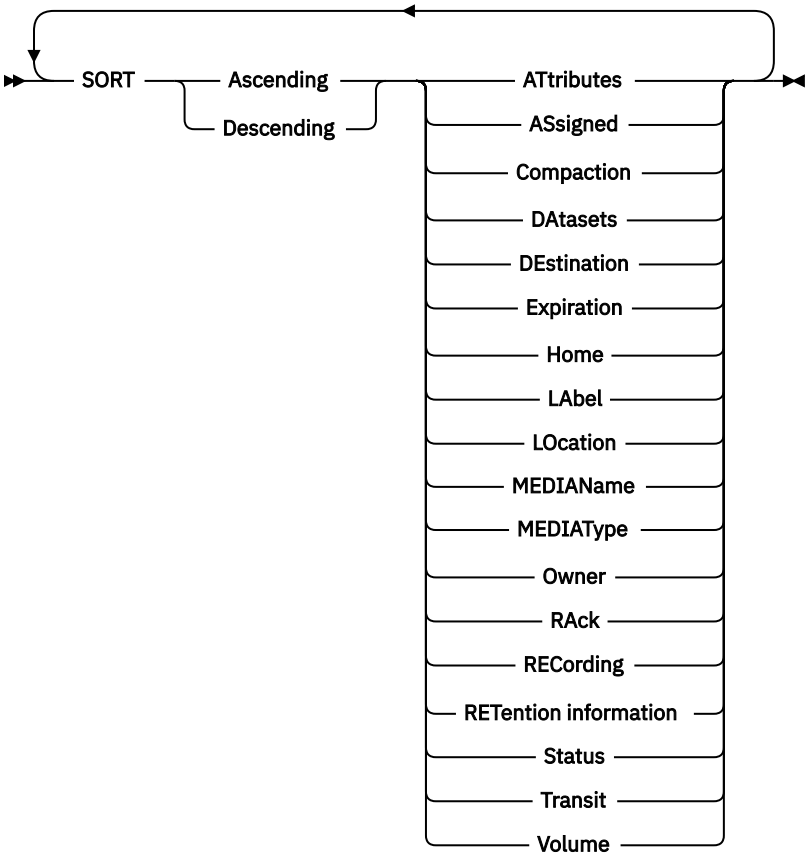


Figure 57. SORT command syntax diagram for volume lists

Volume movements and actions lists

Figure 58 on page 141 describes the syntax to sort a list of volume movements and actions.

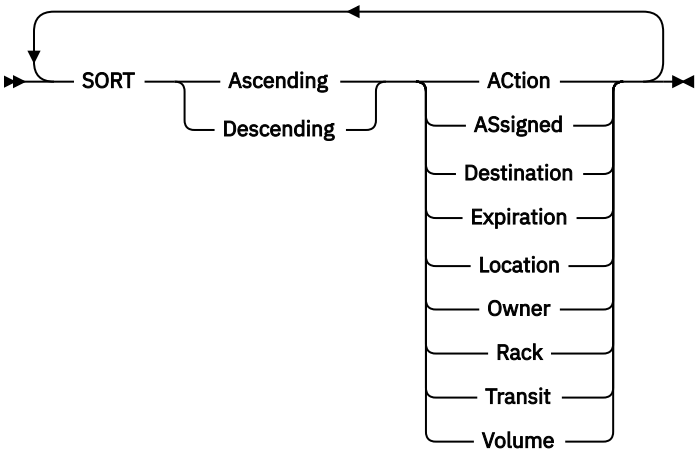


Figure 58. SORT command syntax diagram for lists of volume movements and actions

Software product lists

Figure 59 on page 141 describes the syntax to sort a list of software products.

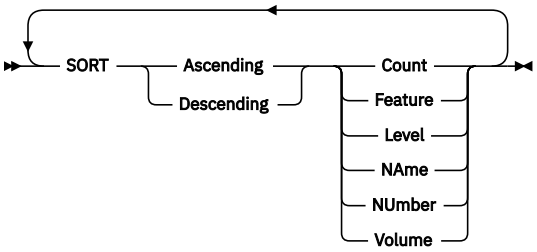


Figure 59. SORT command syntax diagram for software product

Data set lists

Figure 60 on page 141 describes the syntax to sort a list of data sets.

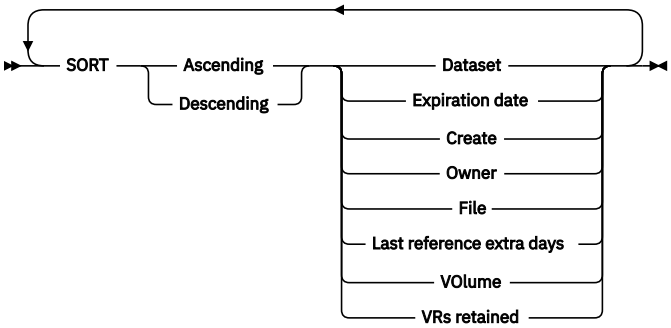


Figure 60. SORT command syntax diagram for data set lists

Vital record specification lists

Figure 61 on page 142 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.

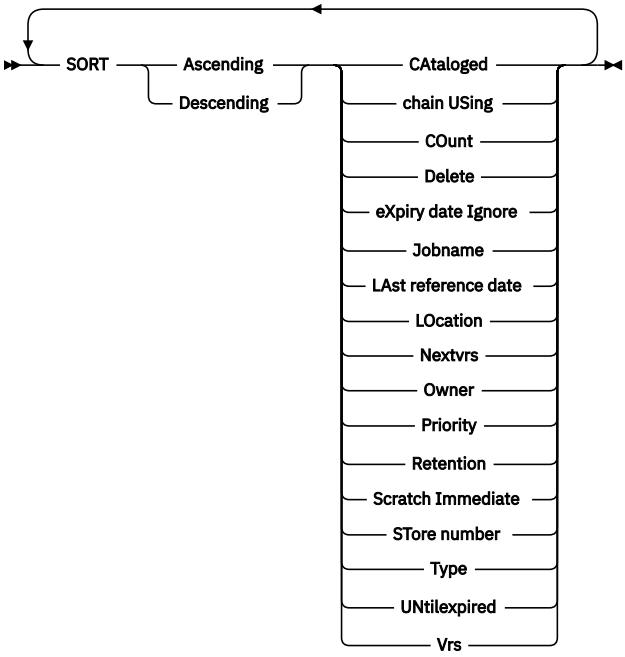


Figure 61. SORT command syntax diagram for vital record specification lists

Report generator lists

Figure 62 on page 142 describes the syntax to sort a list of reports.

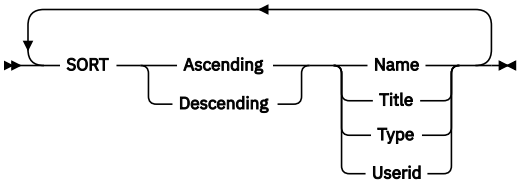


Figure 62. SORT command syntax diagram for report generator lists

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 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 topic 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 380 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 63 on page 143 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 *
  RMM SEARCHVOLUME VOLUME(*) STATUS(SCRATCH) TYPE(PHYSICAL) OWNER(*) LIMIT(100)
  /*
```

Figure 63. Producing a list of scratch volumes

Using the Volume Search panel

You can also use the Volume Search panel, shown in Figure 64 on page 144.

```
EDGPT010                    DFSMSrmm Volume
Search                                                  DFSMSrmm Volume
Search
Command ===>

                                                                More:     +
Volume  . . . . *           May be generic. Leave blank for all volumes
Owner . . . . . *           Owned by a specific user. Default is your userid
Job name  . . .             May be generic
Limit . . . . . *           Limit search to first nnnn volumes
Media name  . .             Limit to a single media name
Vendor  . . . .             Supplier of media
Pool prefix . .             or to a particular pool
Status  . . . .
                            Select volume status. Default is ALL
Dates      Start Date  Start Time  End Date    End Time  Date can be relative
           DD/MM/YYYY  HH:MM:SS    DD/MM/YYYY  HH:MM:SS
Assigned
Expiration
Create . .
Reference
Read . . .
Write  . .
Changed
Moved  . .

Actions . . . .
                            Specify one or more pending actions
Release
  Actions . . .
                            Specify one or more release actions
  Options . . .
                            Specify one or more release options
System use  . .
                            Select system use. Default is ALL
Since . . . . .             Volumes assigned since DD/MM/YYYY
Retention . . .             Volumes retained up to DD/MM/YYYY
Original EXPDT              YES, NO, or a specific date DD/MM/YYYY
Retention Method . .        VRSEL, EXPDT or blank for all
Hold  . . . . .             YES or NO
EDM . . . . . .             YES or NO
Clist . . . . .             YES to create a data set, or NO, or blank


EXPDT Retention method specific:
    RetainBy . . . .                VOLUME, SET, FIRSTFILE, or SETORFIRSTFILE
    Catalog Retained                K-KeptByCatalog N-No U-UntilExpired

Home  . . . . .             Limit to volumes with this home location name
Location  . . .             Limit to volumes by location. May be generic
In container                Stacked volser

Volume type . .             ( LOGICAL , PHYSICAL or STACKED )
MedInf  . . . .             Media information name ( IBM or MEDINF defined)
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 )
Worm  . . . . .             YES or NO to limit volumes based on WORM, or blank

Compaction  . .             Limit to volume
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 )
Volume set  . .             Volumes in a set/chain ( YES, NO, FIRST, or LAST )
Set retained                Volumes retained BYSET ( YES or NO )
Force . . . . .             Volumes used with FORCE ( YES or NO )
```

Figure 64. DFSMSrmm Volume Search panel

DFSMSrmm displays a list of scratch volumes that meet the specified date criteria. 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 380 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 65 on page 145 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 65. 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 95 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 327 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 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 15 on page 120 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 95 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 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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST(''*suffix_string*')).

The data set used for the CLIST output by default will have variable length records and a maximum logical record size of 255 unless you pre-allocate the data set and specify a different record format and LRECL. CLIST data set records can be either fixed or variable length and can optionally be blocked.

When you use a variable record length, the record length includes a 4-byte length field at the front of the record. DFSMSrmm honors the existing DCB attributes of the CLIST data set. When DFSMSrmm allocates a new CLIST data set or you do not specify DCB attributes, the defaults are variable length and an LRECL of 255. If the length of the output record exceeds the LRECL, DFSMSrmm splits the record into multiple records and adds a continuation character, +, to all but the last record.

The LRECL at least long enough to contain the CLIST information for the record type you are searching. In addition, if a suffix or prefix is specified, the minimum LRECL is increased by 1 to allow for the continuation character. If the record is too short, DFSMSrmm increases the record to the minimum size required to hold the output information. The maximum LRECL supported is 32760.

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 you pre-allocate the RMMCLIST data set or the 'prefix.EXEC.RMM.CLIST', they can be placed in an extended addressing space (EAS). DFSMSrmm does not specify EATTR when it dynamically creates the 'prefix.EXEC.RMM.CLIST', which means that EAS will not be used for that data set unless you pre-allocate it.

Operands START and ADD are provided on all SEARCH subcommands to allow records to be added to an existing CLIST data set instead of replacing any existing records in that CLIST file. The RMMCLIST/clist data set is opened either for output or for extend processing depending on the users subcommand request and the DISPosition of the allocated RMMCLIST data set. DISP=MOD overrides the START operand. For example,

```
RMM SV VOLUME(*) CLIST NOLIST START
```

START
Specify this operand to request that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

ADD
Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

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. On the Search panel specify 'YES' in the CLIST data entry field, and press ENTER.
2. On the DFSMSrmm CLIST processing panel:
   a. Enter the RMM TSO command prefix and suffix information, CLIST data set name, CLIST data set size, whether you want to extend existing CLIST processing, and whether you want to view the search results list.
   b. Press ENTER. DFSMSrmm optionally displays the search results and then the DFSMSrmm CLIST Menu panel.
   c. 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 Vol I* for more information.

ISPF data set list utility support

By default, the ISPF Data Set List Utility does not support line commands for tape data sets. You can now configure ISPF to enable DFSMSrmm to support a limited set of the available line commands for tape data sets. Use the ISPF Configuration Utility to customize the ISPF Configuration Table. To enable this new function, select the "Enable RM/Tape Commands" option. For details of how to use the ISPF Configuration Utility, refer to *z/OS ISPF Planning and Customizing*.

The line commands initially supported by DFSMSrmm are I, S, M, and D. These are supported by DFSMSrmm as follows:

- I displays a search results list showing all data sets in the multivolume set for the selected data set.
- S displays the individual data set details. DFSMSrmm determines the first file on the selected volume that matches the selected data set. If other data sets of the same name exist on the volume, the wrong details may be displayed. In that case, use the M line command and then the DFSMSrmm I line command from that results list.
- M displays a search results list showing all data sets on the same volume as the selected data set.
- D releases the volume. If the volume is part of a multivolume set, there is the option to release all volumes in the set.

Once you see the DFSMSrmm results, you are in the DFSMSrmm ISPF dialog and can use any of the available functions including fastpath commands. You can use the DFSMSrmm support either by enabling

the support and using the DSLIST line commands, or you can use the EDGRPD34 exec as a command directly in the DSLIST results. For example:

```
EDGRPD34 I
```

This command processes the DSLIST I line command.

You can also use one of the exec aliases. The exec aliases shipped with DFSMSrmm are RMMI and TI, or you can rename the EDGRPD34 exec to any value you wish. However you choose to run the command outside of the ISPF DSLIST built-in capability, the exec expects that optionally the first parameter may be a line command such as I, S, M or D.

If you do not specify an optional line command when the EDGRPD34 exec is used, the default line command is I. When any other exec or alias name is used, the exec or alias name is used as the line command.

You can customize how the ISPF Data Set List line commands are handled by DFSMSrmm by modifying the EDGRMAIN exec to apply any of the DFSMSrmm results to any of the line commands, and in addition, there are 2 more options:

- Display the volume details for the selected entry
- Display the search results for all the volumes in the same set as the selected entry

ISMF mountable tape volume list support

The EDGRPD34 exec can be used from the ISMF Mountable Tape Volume List. You can also use one of the exec aliases. The exec aliases shipped with DFSMSrmm are RMMI and TI, or you can rename the EDGRPD34 exec to any value you wish. When used from this environment, the information displayed or listed is based on the volume serial number. The EDGRPD34 exec expects that optionally the first parameter is a line command such as I, S, M, or D.

- I displays the multivolume set.
- S displays the volume details.
- M displays a search results list showing all data sets on the same volume.
- D releases the volume. If the volume is part of a multivolume set, there is the option to release all volumes in the set.

Time zone considerations

When you use the DFSMSrmm ISPF dialog, by default, you work in the local system's time. You can also use the built-in capabilities of the dialog to view resources and resource lists in any selected time zone offset. Using time zone within the DFSMSrmm ISPF dialog enables you to view the resources relative to another system in your RMMplex that runs on a different local time as your system, or perhaps volumes are used off the z/OS platform and the other platform runs in a different time zone.

All displayed information is shown in your selected time zone. Any entered dates (through adding or changing) are assumed by DFSMSrmm to also be in the time zone you selected.

You can use the DFSMSrmm Dialog User Options panel to set the selected time zone. Figure 6 on page 6 shows the DFSMSrmm Dialog User Options panel that DFSMSrmm displays. To save time after that, a fast path command allows you to select the time zone used for your dialog sessions without navigating to the DFSMSrmm Dialog User Options panel and using the Time Zone field. The format of the fast path command is:

```
TZ [standard {+/-}HH[:MM[:SS]]] where:
```

- Standard is a specified character string to represent your current time zone to you. There are no rules other than it from 1 to 4 characters. Recommendation is to use the commonly used values such as BST, CET, PDT. DFSMSrmm uses the value only to display back to you when requested.

- + is used for offsets ahead (east of UT) and - is used for offsets behind (west of UT).
- HH is hours
- MM is minutes
- SS is seconds
- An optional colon (:) separates hours from optional minutes and optional seconds.

To revert to local time, issue the fastpath command:

```
TZ LOCAL
```

If TZ is issued with no operands, the current setting is displayed. For example, you could see 'TZ CET+01:00:00'. Also, only data retrieved after the fast path command is issued is returned using the requested time zone.

Figure 66 on page 150 shows the DFSMSrmm Time Zone help panel that DFSMSrmm displays if you require additional help.

```
Panel  Help
EDGH@O19 ------------------------ TIME ZONE -----------------------------HELP
COMMAND ===>

   The current time zone is set to CET  +01:00:00
   The format is                  zone ohh:mm:ss or      LOCAL
      zone      - 1 to 4 character string to represent the time zone to you
      o
           +    - the offset is East, or ahead, of universal time
           -    - the offset is West, or behind, universal time
      hh:mm:ss
           hh   - The offset in hours
           mm   - Minutes are optional. Specify if required
           ss   - Seconds are optional. If specified also specify minutes
      LOCAL     - No selection is made. Local system time is used

   All dates and times displayed or entered in the DFSMSrmm dialog are values
   local to this time zone. To change the time zone you must specify an offset
   value and a text string to identify that zone to you. The offset value is a
   time that your selected time zone is ahead of or behind universal time
   (UTC/GMT). Changes you make only affect future dialog actions and displays.

   For example, to select MST you might enter ==>MST -07
  Use ENTER to continue, END to exit Help.
```

Figure 66. DFSMSrmm Time Zone Help panel

Chapter 9. DFSMSrmm operator procedures

This topic contains information for:

- Operators who might need to stop and restart the DFSMSrmm subsystem, respond to operator messages, manually erase and initialize tape volumes, display status, hold, release, or cancel long running tasks.
- 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 start or restart the DFSMSrmm subsystem shown in Figure 67 on page 151.

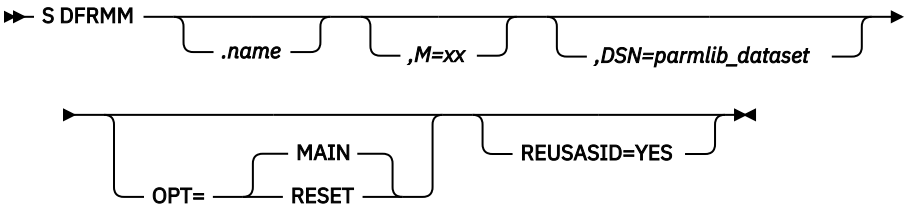


Figure 67. START command syntax diagram

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) 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 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, 2022

RESET
Use OPT=RESET to reset the DFSMSrmm subsystem interface to remove DFSMSrmm from the system on which this command is issued. To completely remove DFSMSrmm also stop the DFSMSrmm subsystem task by issuing: P DFRMM.

REUSASID=YES
When REUSASID=YES is specified on the START command and REUSASID(YES) is specified in the DIAG*xx* parmlib member, a reusable ASID is assigned to the address space created by the START command. If REUSASID=YES is not specified on the START command or REUSASID(NO) is specified in DIAG*xx*, an ordinary ASID is assigned. For more information about reusing ASIDs, see *z/OS MVS Programming: Extended Addressability Guide*.

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 perform DFSMSrmm tasks from an operator console as shown Figure 68 on page 153.

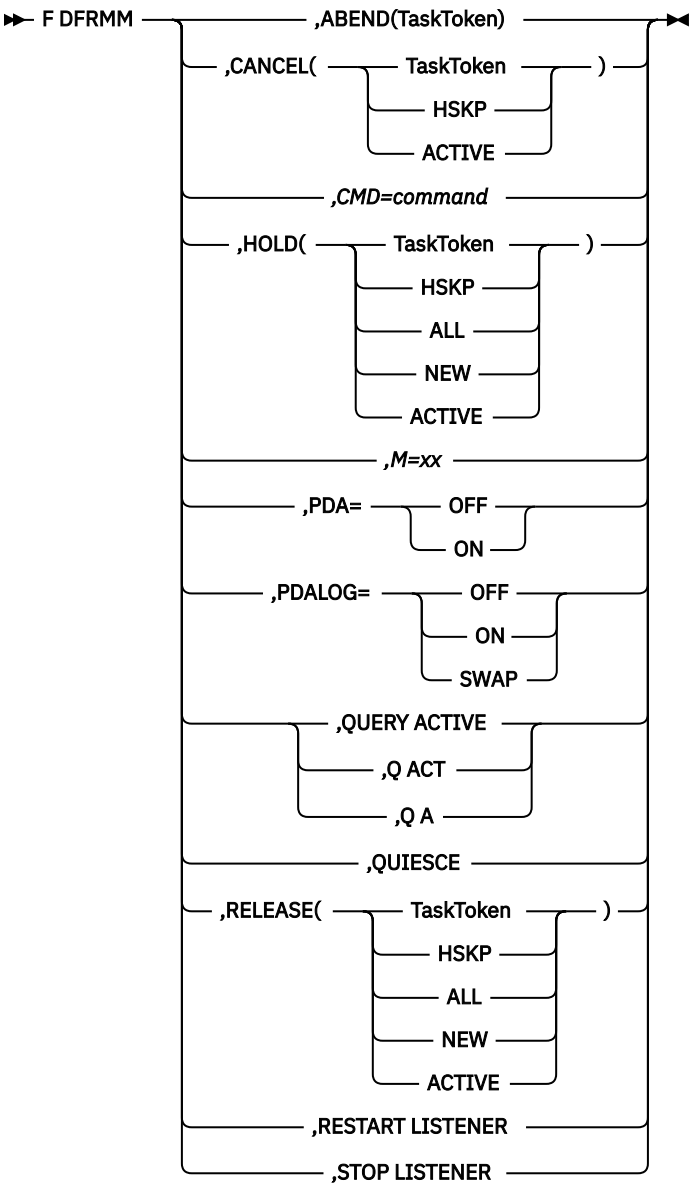


Figure 68. MODIFY command syntax diagram

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 using the `F DFRMM,QUERY ACTIVE` command or one of the other methods listed in "Task management" on page 156.

,CANCEL(TaskToken|HSKP|ACTIVE)
Specifies that you want to interrupt a long running local task. The request ends early and indicates, by a function specific return code and possibly an information message, that processing is incomplete.

TaskToken
A token from the system on which the local task is running that identifies the subtask handling the request. You can obtain the *TaskToken* by using the `F DFRMM,QUERY ACTIVE` command or one of the other methods listed in "Task management" on page 156.

HSKP
DFSMSrmm finds the first available local task running Inventory Management and processes the CANCEL operation. The first available HSKP task, if any, can be displayed with the `F DFRMM,QUERY ACTIVE` command or one of the other methods listed in "Task management" on page 156.

ACTIVE
DFSMSrmm informs all local tasks to process the CANCEL operation. ACTIVE only affects the long running currently active subsystem requests.

,CMD=*command*
Used to issue a DFSMSrmm TSO subcommand from the console and to have the command output returned to your console and the system log. In order to issue commands using the MODIFY command, you authorized to issue that command. If you use z/OS Security Server (RACF) to secure your operator console, the user ID of the logged-on operator is used to authorize the command. If you do not use z/OS Security Server (RACF) to secure your console, the user ID of the DFRMM started procedure is used to authorize the command.

To list the summary volume information recorded by DFSMSrmm for a single volume, use this command.

```
F DFRMM,CMD=LV volser
```

,HOLD(TaskToken/HSKP/NEW/ACTIVE/ALL)
Specifies that you want to interrupt a long running local task, but also cause the task to wait until you are ready to continue. The task releases any DFSMSrmm control data set serialization, then checkpoints what progress has been made, and finally waits for the next command. The next command may be to CANCEL or to RELEASE.

When a task is first subject to HOLD processing, message EDG2020I is issued to your address space so that you are aware that a request may be delayed.

Five minutes after a NEW task is held, or after 20 minutes for all other cases, a message is issued to remind you that one or more tasks are held. Use the `F DFRMM,QUERY ACTIVE` command or one of the other methods listed in "Task management" on page 156 to see which tasks are in HOLD status.

When you have tasks in HOLD, you are also in a WAIT, and this may impact other processing in the system. You can release local tasks individually with the RELEASE(TaskToken/HSKP/ACTIVE) command, but to enable new requests to be processed, you must issue the RELEASE(ALL/NEW) command.

TaskToken
A token from the system on which the local task is running that identifies the subtask handling the request. You can obtain the *TaskToken* by using the `F DFRMM,QUERY ACTIVE` command or one of the other methods listed in "Task management" on page 156.

HSKP
DFSMSrmm finds the first available local task running Inventory Management and processes the HOLD operation.

NEW
Prevents DFSMSrmm from processing any new local subsystem requests.

ACTIVE
Holds only the long running currently active local subsystem requests.

ALL
DFSMSrmm informs all local tasks to process the HOLD operation. Interrupts all long running active RMM subsystem request processing and prevent new requests from starting.

HOLD(ALL) is a combination of the HOLD(ACTIVE) and the HOLD(NEW) functions.

,M=*xx*
xx specifies a parmlib member name suffix with which DFSMSrmm 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, or the defaults table in EDGDEFxx. 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 this 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
- Status
- TCP/IP processing task status

Example: Issue this command to obtain information about DFSMSrmm.

```
F DFRMM,QUERY ACTIVE

EDG1119I DFSMSrmm STATUS IS ACTIVE. JOURNAL ENABLED. LISTENER ACTIVE.  152
EDG1120I Function  System  Task   Name  Started   Token  S IP Status
EDG1113I ADD               JOB=RMMUSERS 06:15:27 00400009      :  :
EDG1113I HSKP              JOB=INVMGMTS 05:29:27 00300002H     :  :
EDG1113I ADD      EZU34    JOB=RMMUSERS 06:15:49 0060000B+Re<06:17:09
EDG1113I C/S      EZU34    STC=DFRMM    00:00:00 00700001 Re>06:16:52
EDG1114I LOCAL TASKS   5, ACTIVE   2, SERVER TASKS   2, ACTIVE   2
EDG1122I                 HELD     1                    HELD     0
EDG1118I      0 QUEUED REQUESTS, INCLUDING      0 NOWAIT      0 CATALOG
EDG1123I NEW REQUESTS ARE HELD
EDG1121I DEBUG: DISABLED, PDA TRACE LEVEL: 1-2-3-4- RESERVE:+06:16:45
EDG1101I DFSMSrmm MODIFY COMMAND ACCEPTED
```

This example shows a local task running HSKP that is subject to HOLD processing. You can also use the abbreviations `Q  ACT` and `Q  A`.

,QUIESCE
Issue this command to manually quiesce DFSMSrmm.

,RELEASE(TaskToken/HSKP/NEW/ACTIVE/ALL)
Specifies that you want to resume processing after a task has been held. For inventory management tasks, the message file will contain messages to show that processing was held and later resumed. For other long running tasks, there may be no indication that processing was temporarily interrupted.

TaskToken
A token from the system on which the task is running that identifies the subtask handling the request. You can obtain the *TaskToken* by using the `F DFRMM,QUERY ACTIVE` command or one of the other methods listed in "Task management" on page 156.

HSKP
DFSMSrmm finds the first available local task running Inventory Management and processes the RELEASE operation. The first available HSKP task, if any, can be displayed using the `F DFRMM,QUERY ACTIVE` command or one of the other methods listed in "Task management" on page 156.

NEW
Allows DFSMSrmm to process any new subsystem requests.

ACTIVE
Releases any active local tasks that were subject to an earlier HOLD command.

ALL
DFSMSrmm informs all local tasks to process the RELEASE operation. Resumes all active DFSMSrmm subsystem request processing and enables new requests to start.

RELEASE(ALL) is a combination of the RELEASE(ACTIVE) and the RELEASE(NEW) functions.

,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.

Task management

When performing DFSMSrmm task management, you can use the:

- QUERY ACTIVE operator command from an operator console at any time to display the status of DFSMSrmm, its local tasks, and request queues.
- TSO subcommand RMM LISTCONTROL STATUS from TSO or with any of the DFSMSrmm APIs when DFSMSrmm is active and able to process subsystem requests.
- CONTROL STATUS ISPF dialog (fast path command, Selection 4.8.12, or Selection 5.7.12 from a TSO ISPF session to interactively display status and manage active tasks when DFSMSrmm is active and able to process subsystem requests

Any task in the system that requests DFSMSrmm subsystem services and fails, or is interrupted because a TSO-user used Attention (ATTN), or is cancelled by the operator, results in any corresponding long-running subsystem request failing. In addition, there are checkpoints built into long-running requests so that when the requestor ends (such as a job being cancelled), DFSMSrmm processing is interrupted at a safe and convenient point. Long running local tasks are DFSMSrmm subsystem requests that last long enough for a task token to be obtained and used.

If the requester is inventory management, the results of the partial processing are available in the MESSAGE file. Long running tasks that support interruption are:

- EDGHSKP inventory management, VRSEL, DSTORE, EXPROC, RPTEXT, and CATSYNCH.

  Note: EDGHSKP ends with return code 12 when cancelled.
- SEARCH *xx* subcommands.

Note: These end with return code 4, reason code 16 when cancelled.

- EDGINERS when building lists of volumes to process.

  Note: EDGINERS processing is still attempted even though one or more search requests of the DFSMSrmm control data set may have been cancelled by the operator. Also, cancelling a task that is processing on behalf of EDGINERS does not cause EDGINERS to be cancelled. To cancel EDGINERS processing, you have to cancel the batch job.

- ADD *xx* and DELETE *xx* subcommands with COUNT specified.

  Note: These end with return code 4, reason code 12 when cancelled.

All these long running tasks run within the DFRMM address space and have checkpoints in their processing that determine whether they should be held, released, or ended. These checkpoints include those that relate directly to the use of the MAXHOLD parmlib option. For example, each time that the MAXHOLD value determines that control data set serialization given up and re-established, there is a time period during which HOLD and CANCEL are effective.

Task management applies only to local tasks. This includes all processing on a standard system, local tasks only on a server system, and all local tasks on a client system. To hold or interrupt a task on a server that is a server task, you have to go to the client system and issue the command there. The QUERY ACTIVE operator command, LISTCONTROL STATUS subcommand, and CONTROL STATUS dialog all help you determine the client that originated a request, and therefore, on which client system any task management commands should be entered. For example, see Figure 69 on page 157 for the results from a QUERY ACTIVE command (the output of the LISTCONTROL STATUS subcommand or CONTROL STATUS dialog would be very similar).

```
EDG1119I DFSMSrmm STATUS IS ACTIVE. JOURNAL DISABLED.   LISTENER ACTIVE.
EDG1120I Function  System  Task   Name  Started   Token  S IP Status
EDG1113I HSKP     EZU161  JOB=RMMUSERK 04:53:28 00C00001 Re<04:53:35
EDG1114I LOCAL TASKS  10, ACTIVE   0, SERVER TASKS  10, ACTIVE   1
EDG1122I                  HELD     0                    HELD     0
EDG1118I      0 QUEUED REQUESTS, INCLUDING      0 NOWAIT      0 CATALOG
EDG1121I DEBUG: DISABLED, PDA TRACE LEVEL: 1-2-3-4- RESERVE: 04:54:11
EDG1101I DFSMSrmm MODIFY COMMAND ACCEPTED
```

Figure 69. Output from QUERY ACTIVE command

In Figure 69 on page 157, the HSKP task is a request running on behalf of system EZU161. To HOLD the task, enter this operator command on the system EZU161 operator console:

```
F DFRMM,HOLD(HSKP)
```

You can use a token instead of the HSKP value. You might need to do this if there are multiple HSKP tasks, and you want to be sure to affect the correct task. You also need to use a token for tasks other than HSKP. To use a token, first issue the Q A command on the EZU161 system. Take note of the token for the task to be affected and then use the token with the operator command on the client system.

Task management for DFSMSrmm TSO subcommands

These DFSMSrmm TSO subcommands can be held, released, or cancelled:

- ADD subcommands with COUNT specified
- DELETE subcommands with COUNT specified
- SEARCH subcommands

Your ability to hold or cancel a task depends on how long the DFSMSrmm subsystem processing takes, depends on how many records are retrieved from the DFSMSrmm control data set before a match is found, and the value used in the MAXHOLD parmlib option. When many DFSMSrmm control data set records are searched, but few meet the search criteria, the chances of using hold or cancel successfully increase. Similarly, for ADD subcommands and DELETE subcommands, the chances for success increases as the number of records to be processed increases.

A TSO user can also interrupt these long-running TSO subcommands by using the Attention (ATTN) key to cause an attention interrupt in the DFSMSrmm command processor. If the DFSMSrmm subsystem is already processing the command, the attention handling is handled in a similar way to the user being cancelled. DFSMSrmm is notified, and if a checkpoint is reached where the subsystem becomes aware the requester has ended, the command processing is interrupted. In order to determine how far processing of the ADD *xx* or the DELETE *xx* subcommands has progressed, use the SEARCH *xx* subcommand.

These task management operator commands are useful during problem, contention, or performance situations, or when you simply want to stop the DFSMSrmm control data set activity from one system.

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 messages to inform you of the delay and to list the job names of the address spaces preventing shutdown. For example:

```
EDG0154I SHUTDOWN OF DFSMSrmm DELAYED BY ANOTHER ADDRESS SPACE
EDG0155I ADDRESS SPACE LIST BY JOBNAME:
         JOBNAME1 JOB2     JOB3    JOB4       JOB5
         NUMBER OF JOBNAMES DELAYING SHUTDOWN =    5
```

Also, DFSMSrmm cannot stop if inventory management is already running. If any long running task prevents DFSMSrmm from stopping, you can use this command to interrupt processing:

```
F DFRMM,CANCEL(TaskToken/HSKP)
```

First, issue the QUERY ACTIVE command to determine the task that is actually preventing the stopping of DFSMSrmm. If any requests are subject to HOLD processing, you must RELEASE or CANCEL them in order for DFSMSrmm to STOP. If you want to end long running tasks in order to STOP DFSMSrmm, issue the CANCEL command. You can use this command to display the status of the tasks:

```
F DFRMM,QUERY ACTIVE
```

You must decide whether to cancel the tasks that have been HELD or to release them. To allow the existing tasks to complete while preventing new tasks starting, issue:

```
F DFRMM,HOLD(NEW)
```

Then to release the tasks that are HELD, issue:

```
F DFRMM,RELEASE(ALL)
```

If you decide to cancel the tasks instead, issue:

```
F DFRMM,CANCEL(ACTIVE)
```

In either case, you can now stop the DFSMSrmm subsystem task by issuing:

```
P DFRMM
```

Except when DFSMSrmm is being shut down for recovery or formal changes to the z/OS 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 does not process 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. When you use the EDGRESET utility, make sure to also stop the DFSMSrmm subsystem task by issuing: P DFRMM.

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 z/OS 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. You can use the command:

```
F DFRMM,CANCEL(TaskToken/HSKP/ACTIVE)
```

to interrupt any long running requests that are currently running. If inventory management is already running, or any tasks are already subject to HOLD processing, the z/OS STOP command is rejected.

Example: To manually quiesce the DFSMSrmm subsystem, use the z/OS MODIFY command:

```
F  DFRMM,QUIESCE
```

When you issue the quiesce command, DFSMSrmm completes any requests being processed. Queued requests 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. If the quiesce command is issued while some local tasks are being held, the quiesce command is rejected, and message EDG1108E is received.

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 z/OS START and PURGE commands:

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

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 z/OS 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.

Some of these messages are WTORs, which require a reply from the operator. The replies for these WTORs can be supplied by system automation, such as the z/OS AUTOR facility. A subset of DFSMSrmm WTORs are defined in the system default parmlib member AUTOR00, and almost all DFSMSrmm WTORs are defined in parmlib members AUTORRM and AUTORRP. Refer to *z/OS DFSMSrmm Implementation and Customization Guide* for more information.

The messages listed in Table 21 on page 160 are identified by type and number.

Table 21. Operator messages

Type of message	Message numbers
Initialization messages. See "Initialization messages" on page 161.	EDG0001I, EDG0101I, EDG0103D, EDG0104E, EDG0105I, EDG0110D, EDG0114I, EDG0115D, EDG0117D, EDG0123D, EDG0204I, EDG0215D, EDG0228E, EDG0238E, EDG0239E, EDG0240E, EDG0241E, EDG0242E, EDG0243I, EDG0353I, EDG0358D, EDG0361D, EDG1001A, EDG1101I, EDG1105I, EDG1106I, EDG1107D, EDG9115I,
Tape mount messages. See "Tape mount messages" on page 165.	EDG4013I, EDG6627A, EDG6628A, EDG6663D, EDGV01D, IEF233A
Tape validation messages. See "Tape validation messages" on page 167.	EDG4021I, EDG4023I, EDG4024I, EDG4025I, EDG4026I, EDG4027I, EDG4028I, EDG4032I, EDG4033I, EDG4035I, EDG4036I, EDG4041I, EDG4065I, EDG4066I, EDG4067I
Tape processing messages. See "Tape processing messages" on page 169.	EDG4000D, EDG4001D, EDG4005E, EDG4006E, EDG4007E, EDG4008A, EDG4010D, EDG4012D, EDG4064D
System error messages. See "System error messages" on page 171.	EDG0154I, EDG0155I, EDG2103D, EDG2104E, EDG2107E, EDG2108E, EDG2110I, EDG2111I, EDG2112I, EDG2113I, EDG2114I, EDG2115I, EDG2116A, EDG9116I

Table 21. Operator messages (continued)

Type of message	Message numbers
Tape Labeling Messages. See "Replying to LABEL procedure messages" on page 178.	EDG6619I, EDG6620I, EDG6621E, EDG6622I, EDG6623I, EDG6624E, EDG6625I, EDG6626A, EDG6627A, EDG6628A, EDG6631I, EDG6642I, EDG6658I, EDG6661E,EDG6662E, EDG6663E, EDG6672E, EDG6677E, EDG6678I, EDG6679I, EDG6680E, EDG6681I, EDG6682I, EDG6683I,

For a complete description of the DFSMSrmm messages, see *z/OS MVS System Messages, Vol 5 (EDG-GLZ)*.

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 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 158 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.

- **EDG0117DDFSMSrmm DYNAMIC INSTALLATION EXITS INITIALIZATION FAILED – REPLY “IGNORE”, “CANCEL” OR “RETRY”**

- **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 z/OS 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.

- **EDG0238E OVERLAPPING VOLUME SET DEFINED FOR *command* TYPE(*type*). *value1* OVERLAPS *value2***
  - A PRTITION or OPENRULE command has specified volume selection operands that conflict with another command.
  - Notify the system programmer. Reply to message EDG0215D as directed. Restart DFSMSrmm when the system programmer has corrected the error.
- **EDG0239E REJECT COMMAND FOUND IN PARMLIB AND NO LONGER SUPPORTED**
  - A REJECT command has been found, but one or more PRTITION and OPENRULE commands are also specified. You must use either REJECT commands or OPENRULE and PRTITION commands.
  - Notify the system programmer. Reply to message EDG0215D as directed. Restart DFSMSrmm when the system programmer has corrected the error.
- **EDG0240E YOU CANNOT MIX IBM AND USER SMF RECORD TYPES FOR SMFAUD AND SMFSEC**
  - Both the SMFAUD and SMFSEC operands have been specified. One of the operands specifies "YES" and the other specifies a SMF record type from the user-assigned range.
  - Notify the system programmer. Reply to message EDG0215D as directed. Restart DFSMSrmm when the system programmer has corrected the error.
- **EDG0241E PRTITION COMMAND TYPE(*type*) *volumeorrange value*LOCATION(*location*) IS NOT A STORAGE HOME LOCATION**
  - The location name you specified does not correspond to a LOCDEF defined location of TYPE(STORAGE,HOME).
  - Notify the system programmer. Reply to message EDG0215D as directed. Restart DFSMSrmm when the system programmer has corrected the error.
- **EDG0242E DUPLICATE MEDINF COMMAND FOR NAME *name* MEDIATYPE *type_id,type_name* RECORDINGFORMAT *format_id,format_name***
  - You have specified duplicate MEDINF commands.
  - Notify the system programmer. Reply to message EDG0215D as directed. Restart DFSMSrmm when the system programmer has corrected the error.
- **EDG0243I*operand* IGNORED FOR SYNONYM MEDINF COMMAND FOR NAME *name* MEDIATYPE *type_id,type_name* RECORDINGFORMAT *format_id,format_name***
  - You have specified CAPACITY or REPLACE operands on MEDINF commands that specify synonym names for MEDIATYPE or RECORDINGFORMAT. The CAPACITY and REPLACE operands do not apply to synonym entries and these operands are ignored.
  - Notify the system programmer.
- **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.
- **EDG1101I DFSMSrmm** *command* **COMMAND ACCEPTED**
  - A stop or modify command was issued and completed. You do not need to reply to this message.
- **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 22 on page 164.

Table 22. 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 eitherRACK= or P00L=. When the rack number or pool identifier is added anywhere else in the message, DFSMSrmm does not add the descriptive text.

These 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 23 on page 166.

Table 23. 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.

    This 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 a system completion code, such as x13, x14, or x37. 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 may be rejected, depending on the installation options. When PRTITION or OPENRULE are used, a volume is rejected if the applicable OPENRULE action is REJECT. When REJECT commands are used, an undefined volume is rejected only if the REJECT option in effect is either REJECT ANYUSE(*) or REJECT OUTPUT(*). The job then ends abnormally.

In the case where for a specific mount request of a volume not defined to DFSMSrmm, the mounted volume does not match the requested one, and the applicable OPENRULE option with action IGNORE is used, DFSMSrmm issues a WTOR (EDG4064D) asking the operator to reply either "I" to allow the mounted volume be ignored by DFSMrmm, or "R" to reject it. When "R" is replied to the WTOR, the mounted volume is rejected and the system mount message is reissued to provide an opportunity to mount another volume.

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.

These 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 z/OS 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 that 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 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 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.
- **EDG4058I VOLUME *volser* REJECTED BY INSTALLATION OPENRULE COMMAND REJECT ACTION**
  - The specified volume cannot be used on this system because the volume matches an OPENRULE entry defined for the installation that specifies an action of REJECT.
- **EDG4059I VOLUME *volser* REJECTED - VOLUMES TO BE REPLACED ARE "READ ONLY"**
  - A volume that has the REPLACE release action set is used for output processing. DFSMSrmm does not allow volumes to be written to if they are to be replaced.
- **EDG4060I VOLUME *volser* REJECTED. OPENRULE ACTION IGNORE BUT USE OUTSIDE OF DFSMSrmm CONTROL NOT AUTHORIZED**
  - The volume matched to an OPENRULE entry that specified that DFSMSrmm ignore this volume. The user authorized to request that the specified volume be ignored, but the user was not authorized, and so the request failed.
- **EDG4065I VOLUME *volser* REJECTED AS REQUESTED BY OPERATOR'S REPLY "R" TO MESSAGE EDG4064D**
  - The operator replied "R" to message EDG4064D to reject the currently mounted volume and get an opportunity to mount another volume.
- **EDG4066I VOLUME *volser* REJECTED. DATA SET BLOCKED BY DSNEXPIRE. IT IS EXPIRED AND NOT RETAINED BY A VITAL RECORD SPECIFICATION.**
  - The DSNEXPIRE(BLOCK) option is in effect. Reading or writing data sets that are not retained and the expiration date for which has passed is prohibited.
- **EDG4067I VOLUME *volser* REJECTED. DATA SET BLOCKED BY FORCEEXPIRE.**
  - The data set being accessed was modified using the CHANGEDATASET FORCEEXPIRE DFSMSrmm subcommand. Reading or writing such datasets is prohibited.

Tape processing messages

These 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 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 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_name* **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 151 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.
- **EDG4064D OPENRULE ACTION IGNORE FOR REQUESTED VOLUME** *req_volser* **, VOLUME** *mnt_volser* **MOUNTED. REPLY "I" TO IGNORE, "R" TO REJECT.**

  For a specific mount request of an undefined volume, DFSMSrmm found that the mounted volume does not match the requested, and the applicable OPENRULE option with action IGNORE is used.
  - Reply "I" if you want to allow the currently mounted volume be ignored.
  - Reply "R" if you want to reject the currently mounted volume and get an opportunity to mount another volume.

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.

These are system error messages for which you might want to set up handling procedures:

- **EDG0154I SHUTDOWN OF DFSMSRMM DELAYED BY ANOTHER ADDRESS SPACE**
  - Look at following message EDG0155I to see the list of jobnames delaying shutdown.
  - Reply to any outstanding WTORs for the jobnames 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 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.

Initializing, erasing, and scanning tape volumes manually

Use the LABEL procedure to run the EDGINERS utility so that you can process initialize, erase, and scan requests manually. With manual processing of EDGINERS, you can initialize and erase, or scan the labels of, any specific tape volume, whether it is already defined to DFSMSrmm or not. Once you initialize a volume, it is defined to DFSMSrmm.

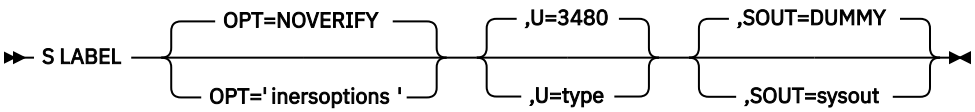
The LABEL procedure is supplied as EDGLABEL in the DFSMSrmm SAMPLIB data set. You must have a tape drive on which you can initialize tape volumes.

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.

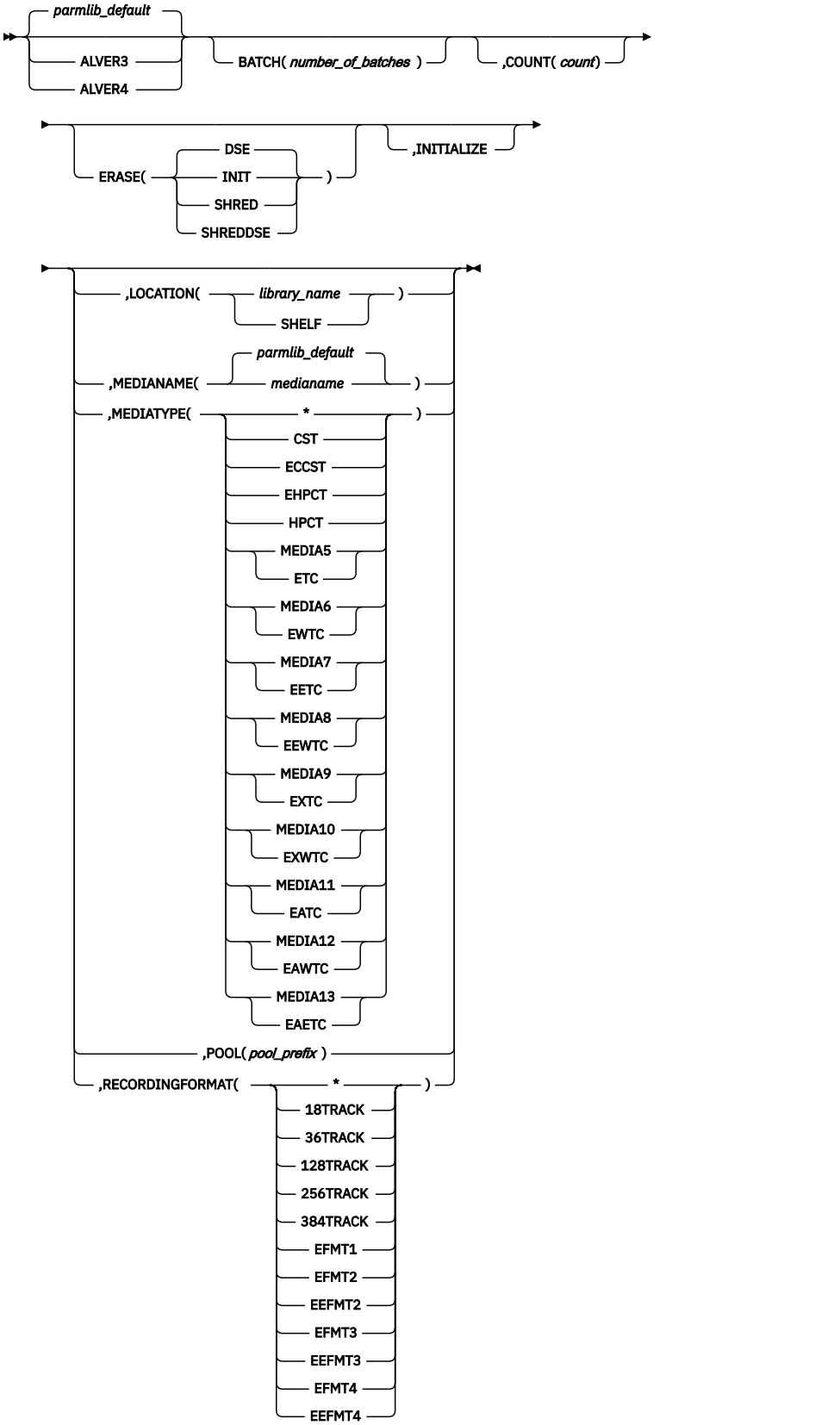
The tape label version for ISO/ANSI version tape labels can be specified in the EDGINERS EXEC JCL. If no tape label is specified, DFSMSrmm uses the version number in the parmlib DEVSUPxx member as the default tape label version when initializing ISO/ANSI tapes. See *z/OS DFSMSrmm Implementation and Customization 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 70 on page 173



inersoptions



where:

OPT=NOVERIFY
This is the default value for OPT. 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 *inersoption*s. 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|ALVER4|*parmlib_default*

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.

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

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(DSE|INIT|SHRED|SHREDDSE)
Use ERASE to request that DFSMSrmm selects 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. When you specify ERASE, DFSMSrmm performs automatic processing.

You can optionally specify an operand value for ERASE to select the action to be performed by the tape drive for the ERASE action. The following values can be specified:

DSE
Specifies that a Data Secure Erase (DSE) should be attempted. This exploits the tape drive hardware capability to erase data from the volume. This is the default for the ERASE operand.

INIT
Specifies that an ERASE action equates to an INIT action; no secure data erase is attempted and the volume is relabelled as if the INIT action had been requested.

SHRED
For encrypted volumes, this value specifies that the Data Key should be made unusable by the drive. For non-encrypted volumes the DSE action is attempted.

SHREDDSE
For encrypted volumes, this value specifies that the Data Key should be made unusable by the drive, and that any non-encrypted residual data on the volume should be subject to DSE. For non-encrypted volumes, the DSE action is attempted.

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* 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(*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 z/OS 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 | MEDIA9 | MEDIA10 | MEDIA11 | MEDIA12 | MEDIA13)
Specifies the volume's physical media type. Use MEDIATYPE to specify a subset of volumes for automatic processing. Use one of these 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 Enterprise Tape Cartridge

MEDIA6/EWTC
IBM Enterprise WORM Tape Cartridge 3592

MEDIA7/EETC
IBM Enterprise Economy Tape Cartridge 3592

MEDIA8/EEWTC
IBM Enterprise Economy WORM Tape Cartridge 3592

MEDIA9/EXTC
IBM Enterprise Extended Tape Cartridge 3592

MEDIA10/EXWTC
IBM Enterprise Extended WORM Tape Cartridge 3592

MEDIA11/EATC
IBM Enterprise Advanced Tape Cartridge

MEDIA12/EAWTC
IBM Enterprise Advanced WORM Tape Cartridge

MEDIA13/EAETC
IBM Enterprise Advanced Economy Tape Cartridge

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.

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 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 | EFMT2 | EEFMT2 | EFMT3 | EEFMT3 | EFMT4 | EEFMT4)
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.

EFMT2
Data has been written to the volume in EFMT2 (enterprise format 2) recording format. A recording format of EFMT2 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EEFMT2
Data has been written to the volume in EEFMT2 (enterprise encrypted format 2) recording format. A recording format of EEFMT2 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EFMT3
Data has been written to the volume in EFMT3 (enterprise format 3) recording format. A recording format of EFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EEFMT3
Data has been written to the volume in EEFMT3 (enterprise encrypted format 3) recording format. A recording format of EEFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EFMT4
Data has been written to the volume in EFMT4 (enterprise format 4) recording format. A recording format of EFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) only.

EEFMT4
Data has been written to the volume in EEFMT4 (enterprise encrypted format 4) recording format. A recording format of EEFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) 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|NOVERIFY
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.

NOVERIFY means that no additional verification is needed. NOVERIFY is the default value.

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 these messages requiring your reply or action. See Figure 71 on page 182 for the command syntax for the INIT, SCAN, and ERASE commands used to reply to LABEL procedure messages.

- **EDG6626A SPECIFY VOLUME "INIT" "SCAN" OR "ERASE" COMMAND OR "END"**
  - Respond with either a volume INIT, SCAN, or ERASE command.
  - You might want to modify DFSMSrmm to handle this automatically.
- **EDG6627A M***drive_number***V(***volser***) R(***rack_number***) TO BE***action vol1_volser, label_type*
  - *action* can be one of ERASED, LABELLED, SCANNED, or VERIFIED.
  - 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 24 on page 180.

    Table 24. 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.

- **EDG6670E UNEXPECTED RETURN CODE RC=04 FROM SUBSYSTEM REQUEST**
  - There may be a mismatch between the levels of the DFSMSrmm EDGINERS utility and the DFSMSrmm subsystem.
  - Make sure that the EDGINERS utility and the DFSMSrmm subsystem are on the same level. Refer to *z/OS MVS Using the Subsystem Interface* for the error code explanations provided for the IEFSSREQ macro and take the appropriate action. Restart DFSMSrmm and resubmit the job.
- **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.
- **EDG6677E VOLUME *volser* SCAN FAILED**
  - The scan of the specified volume *volser* failed.
  - If the request was operator initiated, correct the error and retry the request. If the error persists, report it to the system programmer.
- **EDG6678I VOLUME *volser* SCAN SUCCESSFUL**
  - DFSMSrmm successfully scanned the volume.
- **EDG6679I SCAN RESULTS:**
  - DFSMSrmm issues this message containing the results of the scan request.
- **EDG6680E MISMATCH OF VOLUME STATUS - VOLUME *volser* IS ALREADY PRIVATE IN THE TCDB**
  - You are scanning the labels of volume that is either scratch status in DFSMSrmm or is not defined to DFSMSrmm but is SCRATCH status in the TCDB. EDGINERS attempts to change the volume status in the TCDB to PRIVATE so that it can be mounted for processing. The change use attribute request failed because the volume is already PRIVATE in the TCDB.
    - The TCDB volume status and DFSMSrmm status were already not in synch.
    - The volume changed status from SCRATCH to PRIVATE since EDGINERS checked the volume status - perhaps the volume was mounted by the library for a non-specific request and has been used successfully.

    Depending on the cause of the problem, you may want to synchronize the volume status in the TCDB from the status in DFSMSrmm, or skip the scan of the volume because DFSMSrmm and the volume labels should now be in synch because the volume was used for output.
- **EDG6681I SCAN FOR VOLUME *volser* CONTINUES - VOLUME NOT DEFINED TO DFSMSrmm**
  - You are scanning the labels of a volume that is not defined to DFSMSrmm.
- **EDG6682I SCAN RESULTS (TRUNCATED): FOR FULL DETAILS - REFER TO SYSPRINT FILE**
  - DFSMSrmm issues this multi-line message to the operator console containing the results of the scan request.
  - Review the message contents for mismatches between the volume and DFSMSrmm information. In order to see the complete output refer to and review the contents of the SYSPRINT file.
- **EDG6683I MISMATCH ON** *values*
  - DFSMSrmm issues this message to the operator console and SYSPRINT file containing the results of the comparison between tape label contents and the information defined to DFSMSrmm during a scan request.
  - *values* Is a list of the values that are different between the tape labels and the information defined to DFSMSrmm. The values listed are the column headings in the EDG6679I SCAN RESULTS message.
  - Review the message contents for mismatches between the volume and DFSMSrmm information. Review the contents of the SYSPRINT file to see the complete output.

Initializing or erasing tape volumes

Figure 71 on page 182 describes the syntax for replying to prompts for an ERASE or INIT request:

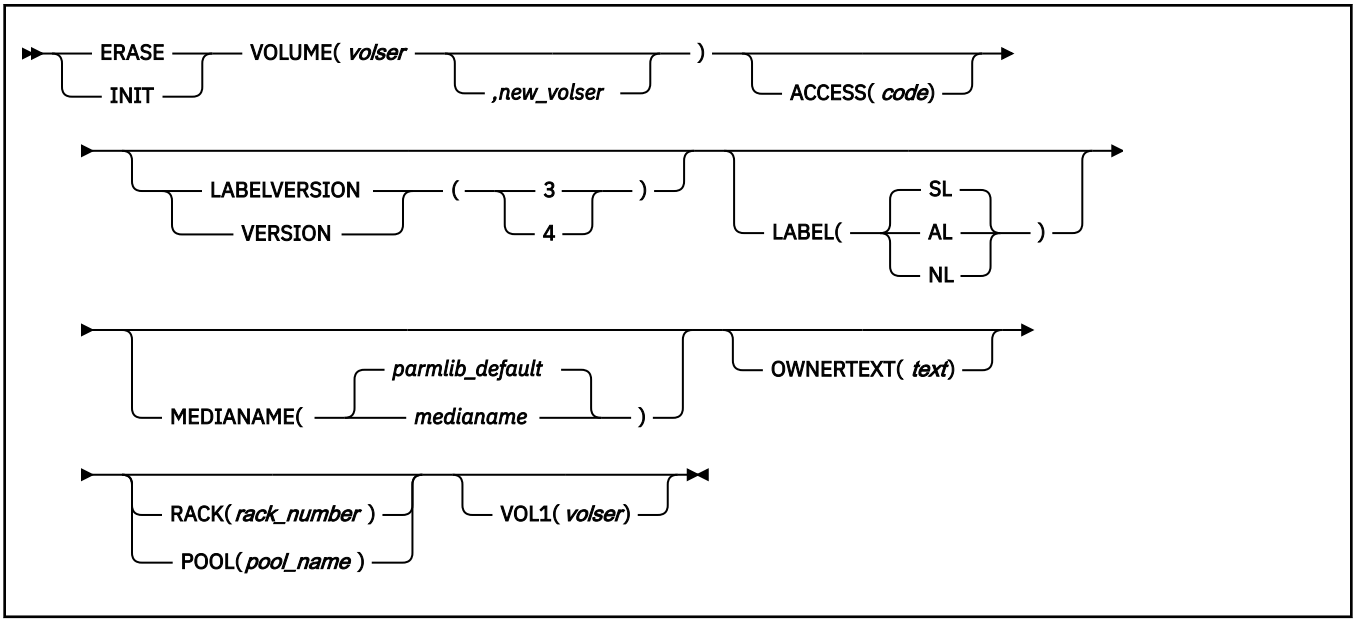


Figure 71. ERASE or INIT request syntax diagram

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

You must specify one of ERASE, INIT, or SCAN.

INIT
Specifies initializing a volume.

You must specify one of ERASE, INIT, or SCAN.

VOLUME(*volser*, *new_volser*)
volser specifies the volume serial number of the volume to be erased or initialized. *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(*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 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 this command format to issue your operator reply:

```
R 21,INIT VOLUME(ABC123) LABEL(AL) POOL(AB*)
```

Scanning a tape volume label

When you need to determine what is on a tape volume, you can use the EDGINERS utility to scan the volume labels. Start the LABEL operator procedure, specifying a tape unit to be used for the request. For example:

```
S LABEL,OPT=NOVERIFY,U=3590-1,SOUT='SYSOUT=*'
```

For system managed volumes that are already defined to DFSMSrmm, you are not required to specify a unit. When EDG6626A WTOR is issued, reply to that with the appropriate SCAN request, such as:

```
R nn,SCAN VOLUME(A00001)
```

A tape mount request is issued for each volume to be scanned. At the end of each SCAN request, review the results displayed with the EDG6682I multi-line message (truncated scan output to console). The complete results of the scan are displayed with message EDG6679I in SYSPRINT file,, including highlighting of any discrepancies found. For volumes defined to DFSMSrmm, check if there are any discrepancies between the tape label information and DFSMSrmm records.

Figure 72 on page 184 describes the syntax for replying to prompts for scanning a tape volume:

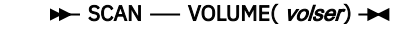


Figure 72. SCAN request syntax

SCAN
Specifies scanning the labels on the volume.

VOLUME
volser specifies the volume serial number of the volume to be scanned. *volser* is one to 6 alphanumeric, national, or special characters. Enclose it in single quotation marks if it contains any special characters.

Handling volume discrepancies

The output of a label scan lists the actual information recorded in the tape labels, which should match that defined to DFSMSrmm. If it does not, the most likely causes are:

- The volume has been written to by another system outside of the control of the DFSMSrmm subsystem on which the SCAN was run.
- The volume has been written to, but the user requested that DFSMSrmm ignore the use of the volume. This could be either through the JCL EXPDT=98000 or through the parmlib OPENRULE settings.

The data on the volume may be valid and need to be retained, perhaps by this or another DFSMSrmm in your environment. You should ensure that the correct DFSMSrmm has a record of, and is managing, the data set and volume. The values displayed for the first file on the volume are a combination of volume and data set attributes. You can use CHANGEVOLUME and ADDDATASET or CHANGEDATASET subcommands to correct the information in the DFSMSrmm CDS. Depending on the data to be corrected,

you may be required to use the FORCE operand on the subcommands when the attributes are recorded by DFSMSrmm during open/close/end of volume processing.

Note: The data defined to DFSMSrmm that appears as a discrepancy may also be valid data that should be retained. This data will be lost after you correct it, so you should consult with the owner of the data before making any corrections to determine whether that data it is still required and, if so, to devise a plan for recreating the data.

Defining volumes

If you use the SCAN function to determine the contents of a volume that is not defined to DFSMSrmm, you can use the details for the first file to add the volume and first file data set information to the DFSMSrmm CDS, using the following commands:

```
RMM AV volser DSN(dsname) ASDATE(crdate) JOBNAME(jobn)
RMM CD dsname VOLUME(volser) JOBNAME(jobn) CRDATE(crdate) LABELNUMBER(dsseq)
       STEP(step) RECFM(recfm) LRECL(lrecl) BLKSIZE(blksize) DEVNUM(device)
```

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 EDG_EXIT100 installation exit. Figure 73 on page 185 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 74 on page 186 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 73. 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 74. Default label format for a round tape

The values for the variables that are shown in Figure 73 on page 185 and Figure 74 on page 186 are:

dsname
The data set name of the file being processed. 1 to 44 characters.

userdata
The user data specified by the 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.

Managing DFSMSrmm installation exits

You can use the RMM LISTCONTROL subcommand to display the status of the DFSMSrmm installation exits.

```
F DFRMM,CMD=LC
```

You should use the D PROG,EXIT,EN=EDG_EXIT* operator command to display a summary of the installation exits used with DFSMSrmm. For the detailed status of a specific installation exit and any defined exit modules use the D PROG,EXIT,EN=EDG_EXITn00, where ‘n’ is currently 1, 2, or 3. Use the SETPROG EXIT operator command to modify the exit recovery handling or to activate and deactivate exit modules.

Chapter 10. Using RMM TSO subcommands

Use the RMM TSO command and a set of subcommands to request DFSMSrmm functions. This topic 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 and provides some additional attribute setting capability than provided by the dialog. You can issue the RMM TSO command and subcommands from within the DFSMSrmm ISPF dialog or outside the dialog, in TSO in the foreground, or in the background by submitting a batch TMP job, or from a TSO CLIST or REXX EXEC, or from System REXX, or from an operator console.

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, but if you are coding commands in procedures, ensure that you spell out the operand completely to avoid future conflicts. Table 25 on page 187 lists the subcommands and their abbreviations.

Table 25. 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
	CHANGEVRS	CS	Change vital record specification 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, 2022

Table 25. 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
	SEARCHOWNER	SO	Create a list of owners
	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
```

Using TSO Subcommands from an operator console

You can issue the RMM TSO command from your operator console and have the command output returned to your operator console and the system log. Example: Use this command format to issue the TSO command from your operator console: F DFRMM,CMD=LV volser This subcommand lists the summary volume information just as if the command had been entered from a TSO session.

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.

This is an example of the JCL required to run a batch TSO TMP, with the DFSMSrmm TSO subcommands, RMM LISTCONTROL and RMM LISTOWNER.

```
//TMP      EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
RMM
```

```
 LISTCONTROL 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*.

Using TSO subcommands from a TSO CLIST or REXX exec

The RMM TSO command is a regular, but authorized TSO command and should function in a TSO environment as an other TSO command. For use from a REXX environment, you must address the TSO environment. For examples of use of RMM subcommands from REXX, see the EDGXMP1 and EDGXMP2 samples in SAMPLIB.

Using RMM TSO subcommands with System REXX

RMM TSO subcommands can be issued by System REXX as long as the target System REXX environment is TSO=YES (No testing has been done with TSO=NO). You should put the REXX exec containing the RMM TSO subcommands in the library supported by System REXX, currently SYS1.SAXREXEC. You can execute any system REXX exec using the operator command "F AXR,rexx_name" from a console. Just as with regular REXX, if your system REXX exec sets the REXX variable SYSAUTH.EDGDATE, the RMM subcommand processing creates REXX variables as expected. If SYSAUTH.EDGDATE is not set, RMM subcommand processing results in no REXX variables being created and the resulting line command output is issued to the console.

For example:

```
/* REXX */
 sysauth.edgdate = 'AMERICAN'
 "RMM SV OWNER(*)"
 say 'command = RMM SV OWNER(*)'
 say 'num vols = ' edg@vol.0
 say 'first vol found = ' edg@vol.1
```

The output on the console:

```
command = RMM SV OWNER(*)
num vols = 10
first vol found = V10001
```

When you use System REXX from the console, it makes a difference whether you are logged on to the console or not.

- Logged on with a specific RACF userid:
  - System REXX creates an ACEE with this userid and for RMM subcommand authorization this ACEE and the appropriate DFSMSrmm related RACF profiles are used
  - RMM uses this userid as the resource owner, if the OWNER operand is not specified in the issued command
  - RMM records this userid as Last Change Userid

    Note: RMM can also record these special last change userids, which are not driven by a subcommand:

    ***CAT**
    Updates due to catalog status changes

    ***HKP**
    Updates due to inventory management

    ***MIM**
    Updates due to SARS MIM message interception

    ***OAM**
    Updates due to system-managed tape support

***OCE**
Updates due to Open/Close/EOV support

***UT**
Updates due to execution of EDGUTIL

***WTO**
Updates due to message interception

- If your SEARCH subcommand results in CLIST processing; without RMMCLIST DD pre-allocated - RMM uses the user prefix (which is initially set to the RACF userid by System REXX) as HLQ for the dynamically allocated CLIST data set with RMMCLIST DD pre-allocated - the specified data set is used. Remember that when you dynamically allocate RMMCLIST DD within system REXX using the ALLOCATE command that the TSO convention for quoted and unquoted data set names applies and if an unquoted data set name is specified the user prefix is used as the HLQ for the data set

- If not logged on :
  - IEESYSAS (the jobname of a common System Address Space procedure) is used as the DFSMSrmm resource owner, Last Change userid and CLIST data set HLQ, instead of the RACF userid, that is used in the logged-on case.
  - RMM subcommand authorization has no ACEE on which to base authorization checking, therefore most commands will fail authorization checking. The exceptions are those where the STGADMIN.EDG profiles specify a universal access, which enables users undefined to RACF to use RMM subcommands.
  - If you attempt to pre-allocate a CLIST data set for the RMMCLIST DD using an unquoted data set name remember that there is no user prefix available, so the results will be as if you specified a quoted data set name. For example, "ALLOCATE DA(SYSREXX.CLIST)...." will result in CLIST data set 'SYSREXX.CLIST'; no prefix is added.

ADDBIN: Adding a bin number in a storage location

Purpose

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 208 for the combined description of the ADDRACK and ADDBIN operands.

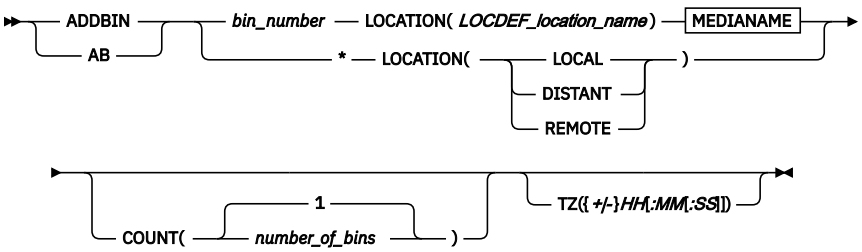
DFSMSrmm defines shelf space in storage locations as bin numbers. Use the ADDBIN subcommand to define empty or available shelf locations in storage locations. The storage locations can be DFSMSrmm built-in storage locations or storage locations that are defined by your installation.

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 327 for more information.

Format



MEDIANAME

Parameters

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.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is*{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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 211 for more information.

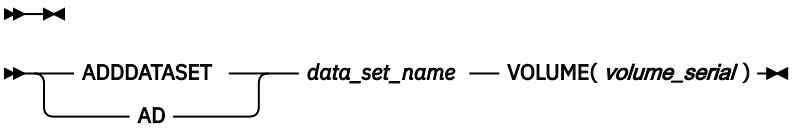
- Define information about data sets preceding the data set you want to add to DFSMSrmm, or DFSMSrmm fails your requests. You can use this subcommand to add information about all preceding data sets on the volume to DFSMSrmm if DFSMSrmm has no record of them.

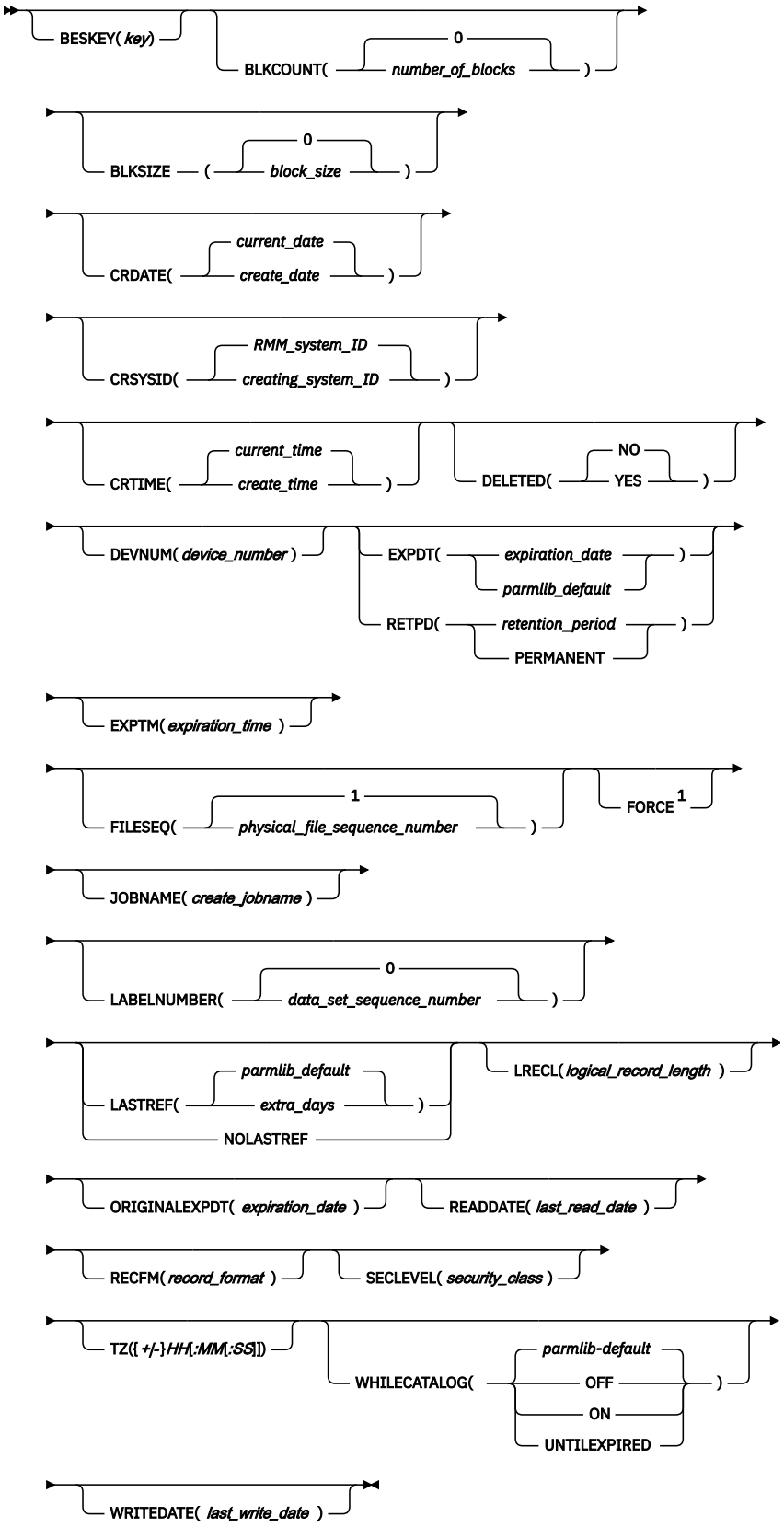
Use the ADDDATASET subcommand to manually define a data set on a volume to DFSMSrmm. The volume on which the data set resides must have either master or user status.

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.

Format





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,

Parameters

BESKEY(*key*)
Use the BESKEY operand to set or change the CA Tape Encryption key index, which is set by the BES subsystem. *key* is an encryption key index and can be expressed as a number from 0 to 2147483647, or a hexadecimal value from X'00' to X'FFFFFFFF'.

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 label. The minimum allowable decimal value is 0; the maximum allowable decimal value is 4294967295. DFSMSrmm uses BLKCOUNT, together with BLKSIZE, to calculate the approximate size of the data set and the sum of all data set sizes to set the volume usage. If you do not use BLKCOUNT, DFSMSrmm cannot list the space used for the data set or volume when you request it. The default value is 0.

Note: The total block count cannot be set or changed by subcommand. It is set based on information recorded during CLOSE processing.

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 and the sum of all data set sizes to set the volume usage. If you do not use BLKSIZE, DFSMSrmm cannot report space usage. The default value is 0.

CRDATE(*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 2012/001. 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 12001. 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.

CRDATE can be abbreviated as DATE.

CRSYSID(*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. 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.

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. You can specify CRSYSID if you are authorized to use the FORCE operand.

CRSYSID can be abbreviated as SYSID.

CRTIME(*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.

CRTIME can be abbreviated as TIME.

data_set_name
Specifies the name of the data set being added.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

DFSMSrmm does not check quoted data set names for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. Data set names without quotes must pass these data set naming rules:

- A data set name can have one or more qualifiers.
- Each qualifier is one to eight characters. The first character alphabetic (A to Z) or national (# @ $). The remaining seven characters can be either alphabetic, numeric (0 - 9), national, or a hyphen (-).
- Qualifiers are separated by a period.
- DFSMSrmm adds your TSO PROFILE PREFIX value as the high-level qualifier.
- The data set name must not include a member name.

This operand is required and must immediately follow the ADDDATASET subcommand.

DELETED(NO | YES)
DELETED(YES) specifies the data set is deleted.

The default value is NO.

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 2012/001. 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 12001. 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.

When you specify the expiration date for a data set record representing one part of a multivolume data set on volumes managed by the EXPDT retention method, and they are retained by SET or VOLUME, DFSMSrmm updates the expiration date and time for all the data set records for the data

set. When you do not specify an expiration date, DFSMSrmm copies the expiration from the preceding data set record in the multivolume data set chain.

A data set on a volume managed by the EXPDT retention method and retained by FIRSTFILE can affect the expiration of volume or multivolume set only if it is the first file of the volume or volume set.

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.

EXPTM(*expiration_time*)
Specifies the time when the data set should be considered for release. The format for *expiration_time* is *hhmmss*, where:

hh
is hours

mm
is minutes

ss
is seconds

The expiration date is determined by the EXPDT or RETPD parameters. The time zone offset specified in the TZ options affects the resulting expiration time.

FILESEQ(*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.

FILESEQ can be abbreviated as SEQ.

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 characters other than blank, comma, and semicolon. You cannot use a generic jobname. Any jobname you use 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.

LASTREF(*extra_days*)
LASTREF(extra_days) specifies the number of days that the data set will be retained after the data set was last referenced.

LASTREF(*extra_days*) applies only to data sets on volumes managed by the EXPDT retention method. LASTREF cannot be specified for data sets on volumes managed by the VRSEL retention method.

extra_days is a decimal number between 0 and 93000. The value must not exceed the maximum retention period MAXRETPD specified in the DFSMSrmm EDGRMMxx parmlib member. An *extra_days* value of 0 has the same effect as using NOLASTREF. DFSMSrmm uses the LASTREF extra days to evaluate the data set expiration date. The extra days are added to the date of last reference. The data

set expiration date is set to the maximum of the date calculated using LASTREF extra days, and the date resulting from applying the EXPDT, RETPD, or default RETPD. Any reference to the data set by a read or write operation will redetermine the expiration date.

When a file is added to a multivolume data set, the LASTREF or NOLASTREF attribute is copied from the preceding file. For a volume set retained by VOLUME or SET DFSMSrmm ensures that the LASTREF(*extra_days*) or NOLASTREF data set attribute is the same for all files of a multivolume data set. For a volume set retained by FIRSTFILE, no additional processing is performed to keep the LASTREF extra days attribute consistent across the multivolume data set, because the expiration date depends only on the LASTREF extra days attribute of the first file of the first volume.

If neither LASTREF nor NOLASTREF is specified for a new data set, DFSMSrmm uses the LASTREF default value specified in the OPTION RETENTIONMETHOD(EXPDT) command in the parmlib member EDGRMMxx

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.

NOLASTREF
NOLASTREF specifies that DFSMSrmm does not consider the data set last reference date when evaluating the data set expiration date.

NOLASTREF applies only to data sets on volumes managed by the EXPDT retention method. NOLASTREF cannot be specified for data sets on volumes managed by the VRSEL retention method.

When a file is added to a multivolume data set, the LASTREF or NOLASTREF attribute is copied from the preceding file. For a volume set retained by VOLUME or SET DFSMSrmm ensures that the LASTREF(*extra_days*) or NOLASTREF data set attribute is the same for all files of a multivolume data set. For a volume set retained by FIRSTFILE, no additional processing is performed to keep the LASTREF extra days attribute consistent across the multivolume data set, because the expiration date depends only on the LASTREF extra days attribute of the first file of the first volume.

If neither NOLASTREF nor LASTREF is specified for a new data set, DFSMSrmm uses the LASTREF default value specified by the OPTION RETENTIONMETHOD(EXPDT) command in the parmlib member EDGRMMxx.

ORIGINALEXPDT(*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 2012/001. 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 12001. 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.

`ORIGINALEXPDT` can be abbreviated as `OEXPDT`.

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 2012/001. 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 12001. 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 you want to set a date in the future, the FORCE operand is required.

If a vital record specification or a data set LASTREF attribute indicates that DFSMSrmm retains 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 ISO/ANSI records

DB
Blocked variable-length ISO/ANSI records

DS
Variable-length ISO/ANSI spanned records

DBS
Variable-length ISO/ANSI blocked spanned records

You can also append either A or M to the fixed and variable formats.

A
The record contains ISO/ANSI 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 ISO/ANSI 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 ISO/ANSI records containing ISO/ANSI 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 decimal number from 0 to 93000, or PERMANENT. The retention period is added to the current date and time to compute the EXPDT and the EXPTM. RETPD(0) causes the default retention period to be used for the dataset as defined in the PARMLIB. RETPD cannot exceed the maximum retention period MAXRETPD specified for your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

When you specify the expiration date for a data set record representing one part of a multivolume data set on volumes managed by the EXPDT retention method and retained by SET or VOLUME, DFSMSrmm updates the expiration date and time for all the data set records for the data set. When you do not specify an expiration date, DFSMSrmm copies the expiration from the preceding data set record in the multivolume data set chain.

Changing the retention period for a data set on a volume managed by the EXPDT retention method and retained by FIRSTFILE has no effect on the expiration of a volume or multivolume set unless it is the first file of the volume or volume set.

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 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 327 for more information. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information on using security class definitions.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

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.

WHILECATALOG(parmlib-default | OFF | ON | UNTILEXPIRED)
Specifies the WHILECATALOG setting for the data set. WHILECATALOG applies only to data sets on volumes managed by the EXPDT retention method.

Specify OFF to indicate that retention of this data set should not depend on its catalog status and that only the expiration date is to be taken into account.

Specify ON to indicate that, if DFSMSrmm sets the data set status to cataloged during CATSYNCH, the data set should not be expired as long as it is cataloged, nor should it be expired prior to the expiration date. Changing the WHILECATALOG setting to ON does not affect the expiration date.

Specify UNTILEXPIRED to indicate that the data set should expire after its status changes to uncataloged or if the expiration date is reached. If CATSYNCH changes the status of the data to cataloged, and it later becomes uncataloged, its expiration date will be decreased, as defined by the CATALOG DAYS parmlib option. If CATSYNCH changes the data set status from Unknown to Uncataloged, its expiration date will not be decreased.

When WHILECATALOG(ON) is specified, the data set expiration date is displayed with **(KeptByCatlg)** appended to the date, if the data set is cataloged. When WHILECATALOG(UNTILEXPIRED) is specified, the data set expiration date is displayed with **(OrUncatlg)** appended to the date, if the data set is cataloged.

Note: An uncatalog event for a data set that has WHILECATALOG(UNTILEXPIRED) specified could cause the expiration date of the data set to decrease. If the data set expiration date decreases and the current volume or volume set expiration date was copied from the expiration date of the data set being uncataloged, DFSMSrmm will assign a new expiration date for the volume or volume set. DFSMSrmm will scan through all of the data sets on the volume or volume set and find the maximal expiration date, which will then be assigned to the volume or volume set.

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 2012/001. 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 12001. 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 want to set a date in the future, the FORCE operand is required.

If a vital record specification or a data set LASTREF attribute indicates that DFSMSrmm retains 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.

Examples

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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

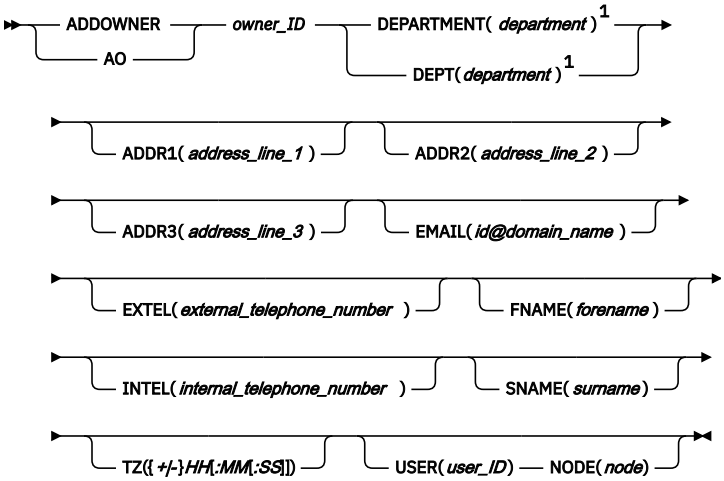
Use the ADDOWNER subcommand to define an owner to DFSMSrmm. An owner can be an individual or a group defined by a RACF group name, or any other value you choose.

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, or a valid e-mail address. DFSMSrmm uses this address to notify the owner when the owner's volumes are eligible for release. Notification used as one of the actions to be performed upon the volume's release, and the parmlib option, OPTION NOTIFY, set to Y. Use the ADDVOLUME or CHANGEVOLUME subcommands to set release actions for a volume. See “ADDVOLUME: Adding volume information” on page 211 or “CHANGEVOLUME: Changing volume information” on page 269 for more information. See *z/OS DFSMSrmm Implementation and Customization Guide* for more information on setting parmlib options.

Format



Notes:

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

Parameters

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.

EMAIL(*id@domain_name*)
Specifies an Internet identifier for the user represented by this owner record. An Internet identifier specifies the path address of the mail recipient. The format of recipient is equivalent to the path syntax, as described in RFC 821, without the (<) and (>) delimiters. The Internet Protocol suite is still evolving through requests for comments (RFC). New protocols are being designed and implemented by researchers and are brought to the attention of the Internet community in the form of RFCs. See *z/OS Communications Server: IP User's Guide and Commands* for details.

This EMAIL operand has one of these formats:

- user_id@host_name.domain - User on a host in a specified domain.
- user_id%nje_host_name@ gateway_name.domain - User on an NJE or RSCS node connected to a TCP network at gateway_name.

There is no default value. When you specify EMAIL, the USER and NODE values are ignored by NOTIFY processing, and any notify messages are sent using your system's SMTP server.

The maximum length of the value supported by DFSMSrmm is 63 characters.

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.

The Owner 'SMTP' is now a reserved owner name value that you can use to configure the Node name and SMTP server address space or machine name. You use the NODE operand to identify the node that runs the SMTP server, and the USER operand to identify the SMTP server. Note: You must specify both values. When you do not have the Owner SMTP defined, DFSMSrmm uses the JES node name of the running system and SMTP as the SMTP server address space name.

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.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is*{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

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.

Examples

Task: Add details for the new owner information shown in Table 26 on page 205.

Table 26. New owner information

Field name	Sample value
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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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.

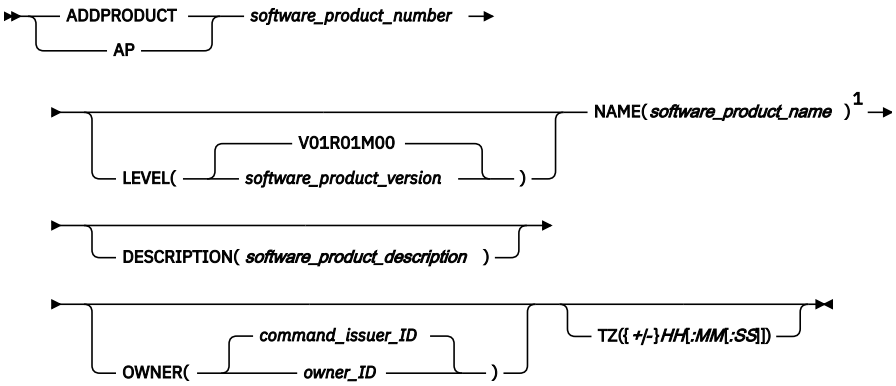
After you have defined the software product to DFSMSrmm, use the ADDVOLUME subcommand to associate one or more volumes with the software product. You must define the software product to DFSMSrmm before you associate one or more volumes with it. The volumes must not be already defined to DFSMSrmm. Use the ADDVOLUME subcommand to define new volumes. See "ADDVOLUME: Adding volume information" on page 211 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 30 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 268 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.

Format



Notes:

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

Parameters

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 alphanumeric or national characters.

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 373 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.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

Examples

Task: Add details of the z/OS software product with product number 5650-ZOS, version 2.1.0.

Command:

```
RMM ADDPRODUCT '5650-ZOS' NAME('z/OS') LEVEL(V02R01M00)
```

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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

This topic describes the combined description for the ADDRACK subcommand and its alias ADDBIN. See "ADDBIN: Adding a bin number in a storage location" on page 190 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 27 on page 208 shows the relationship between rack numbers and the three types of volumes that DFSMSrmm manages.

Table 27. 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.

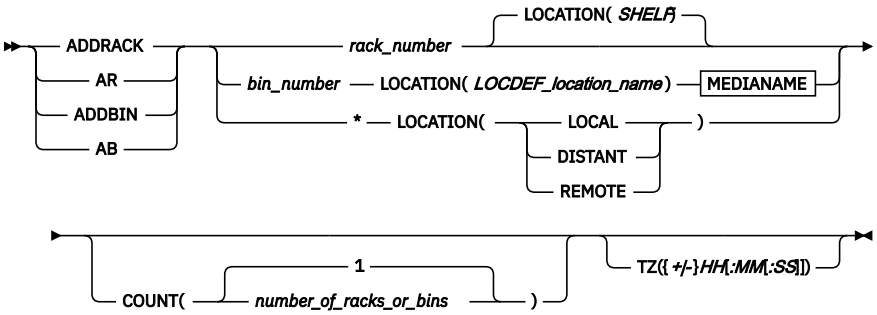
Table 27. Assigning rack numbers to volumes that DFSMSrmm manages (continued)		
For a	**You**	**Optionally**
Stacked Volume	Can specify a rack number or pool prefix.	DFSMSrmm tries to use the volume serial number of the volume as the rack number. If DFSMSrmm cannot find the rack number that matches the 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 327.

See "LISTBIN: Displaying information about a shelf location" on page 325 to obtain information about individual bin numbers.

Format



MEDIANAME

Parameters

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 99999.

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 the same as the volume serial number.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose of ADDVOLUME subcommand

Before you begin: To use the RMM ADDVOLUME subcommand, you need either CONTROL access to the STGADMIN.EDG.MASTER resource profile, or CONTROL access to the STGADMIN.EDG.AV.status.volser profiles, depending on the security roles you have implemented.

Use the ADDVOLUME subcommand to add one or more volumes to DFSMSrmm. Use the MEDINF operand to connect a volume to the media information defined in DFSMSrmm parmlib member EDGRMMxx.

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 28 on page 212.

Table 28. 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.

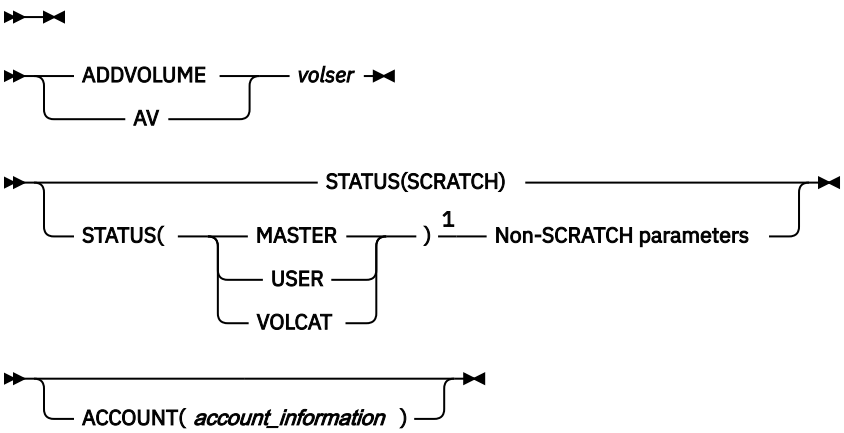
Use the NUMBER, FEATCD, and LEVEL operands to associate the volume with a software product. The software product already defined to DFSMSrmm before you can associate it with a volume.

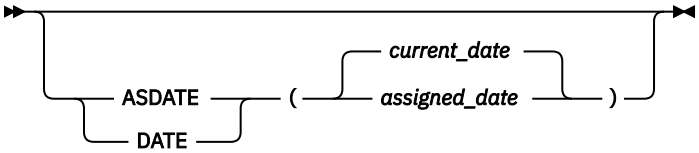
When you specify ADDVOLUME STATUS(VOLCAT), DFSMSrmm uses information in the TCDB to update the DFSMSrmm control data set. The control data set information that can be updated includes: COMPACTION, CONTAINER, EXPDT, HOME, LOCATION, MEDIATYPE, READDATE, RECORDINGFORMAT, SPECIALATTRIBUTES, STATUS, STORAGEGROUP,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

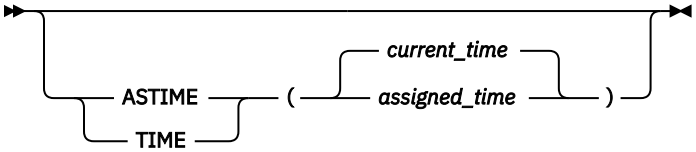
If you use any non-scratch operands when adding scratch volumes, DFSMSrmm ignores the non-scratch operands.

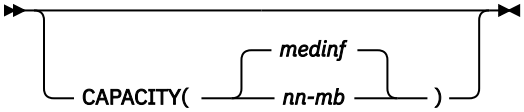
Format of ADDVOLUME volume operands

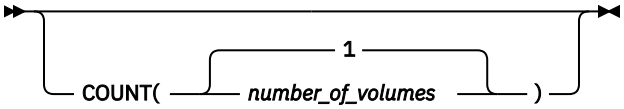
ADDVOLUME Volume Operands

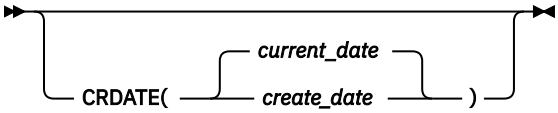


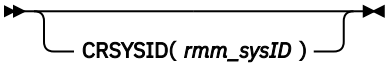

ASDATE
DATE
(current_date assigned_date)

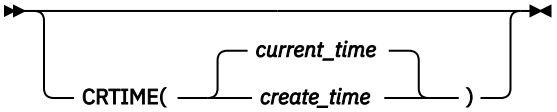

ASTIME
TIME
(current_time assigned_time)

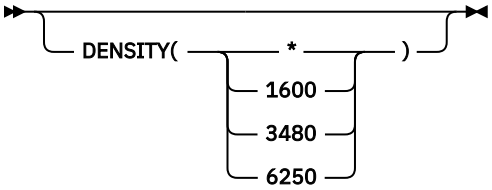

CAPACITY(medinf nn-mb)


COUNT(1 number_of_volumes)

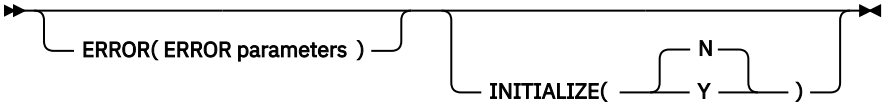

CRDATE(current_date create_date)

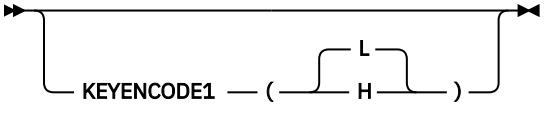

CRSYSID(rmm_sysID)

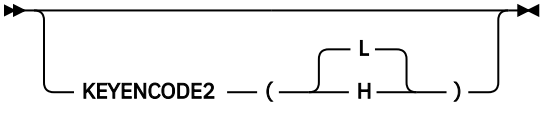

CRTIME(current_time create_time)

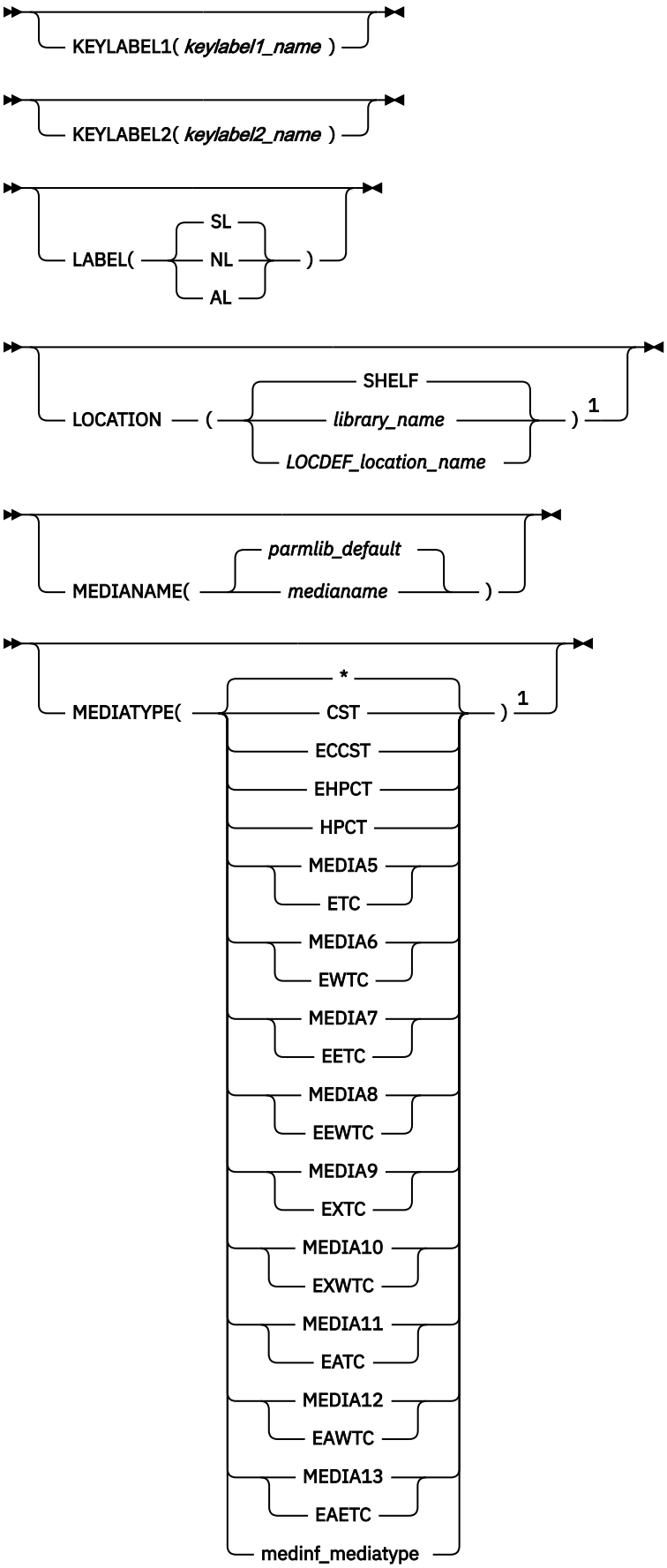

DENSITY(* 1600 3480 6250)

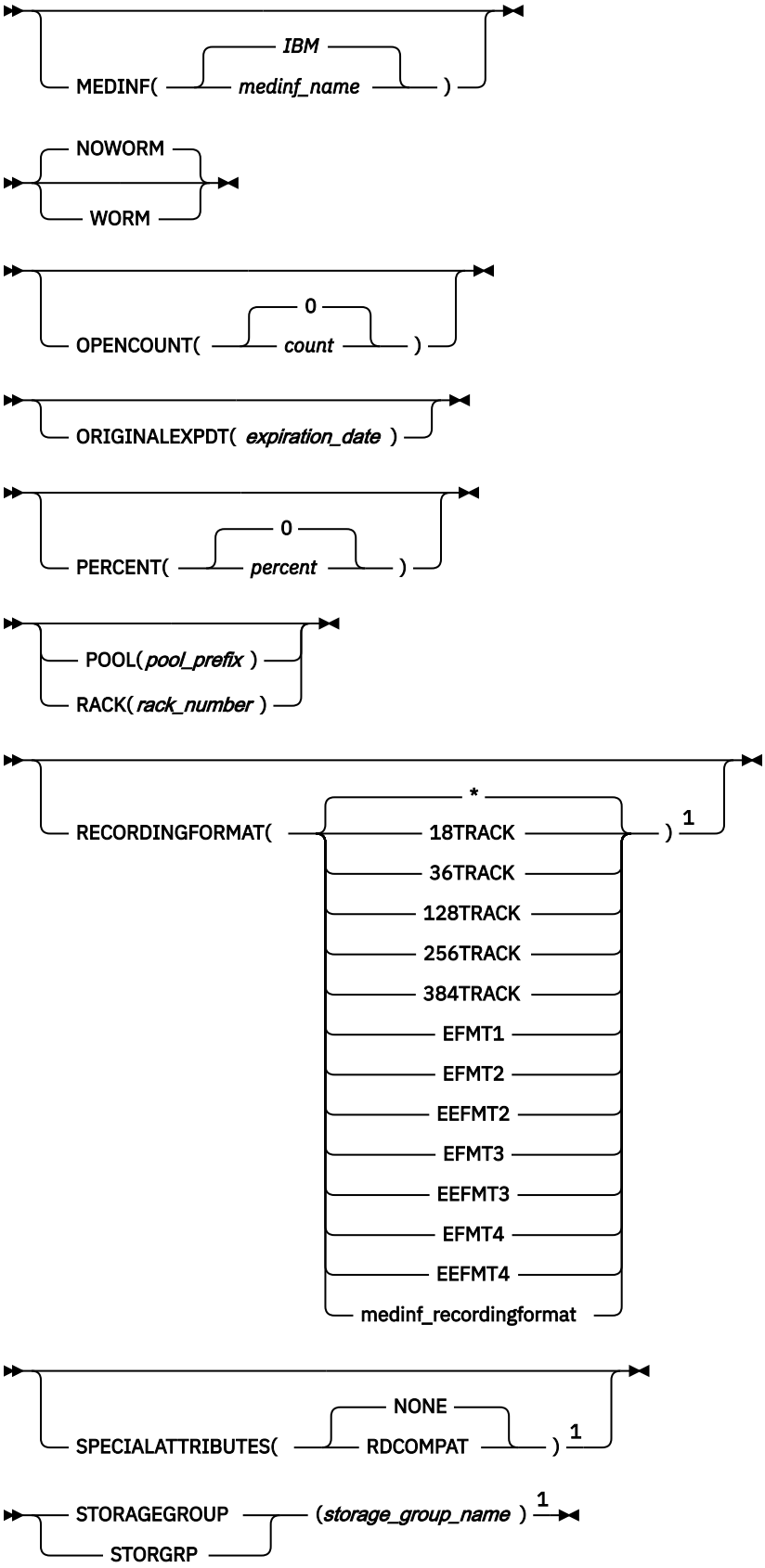
DESCRIPTION(text)

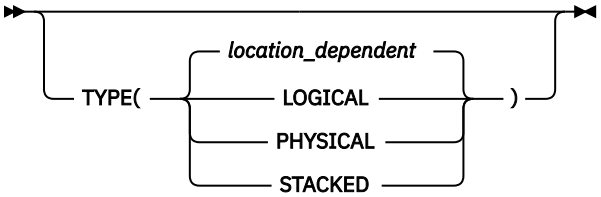

ERROR(ERROR parameters)
INITIALIZE(N Y)

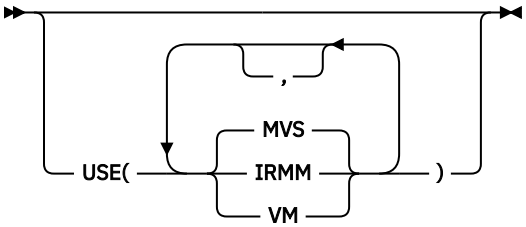

KEYENCODE1 (L H)

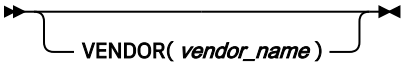

KEYENCODE2 (L H)

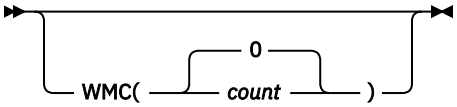

KEYLABEL1(keylabel1_name)
KEYLABEL2(keylabel2_name)
LABEL(SL NL AL)
LOCATION (SHELF library_name LOCDEF_location_name) 1
MEDIANAME(parmlib_default medianame)
MEDIATYPE(* CST ECCST EHPCT HPCT MEDIA5 ETC MEDIA6 EWTC MEDIA7 EETC MEDIA8 EEWTC MEDIA9 EXTC MEDIA10 EXWTC MEDIA11 EATC MEDIA12 EAWTC MEDIA13 EAETC medinf_mediatype) 1

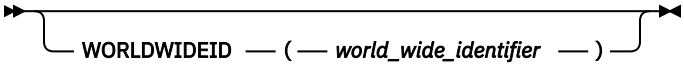

MEDINF(IBM medinf_name)
NOWORM
WORM
OPENCOUNT(0 count)
ORIGINALEXPDT(expiration_date)
PERCENT(0 percent)
POOL(pool_prefix)
RACK(rack_number)
RECORDINGFORMAT(* 18TRACK 36TRACK 128TRACK 256TRACK 384TRACK EFMT1 EFMT2 EEFMT2 EFMT3 EEFMT3 EFMT4 EEFMT4 medinf_recordingformat) 1
SPECIALATTRIBUTES(NONE RDCOMPAT) 1
STORAGEGROUP STORGRP (storage_group_name) 1



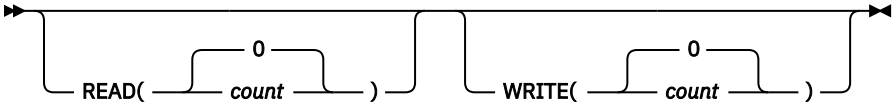








ERROR parameters

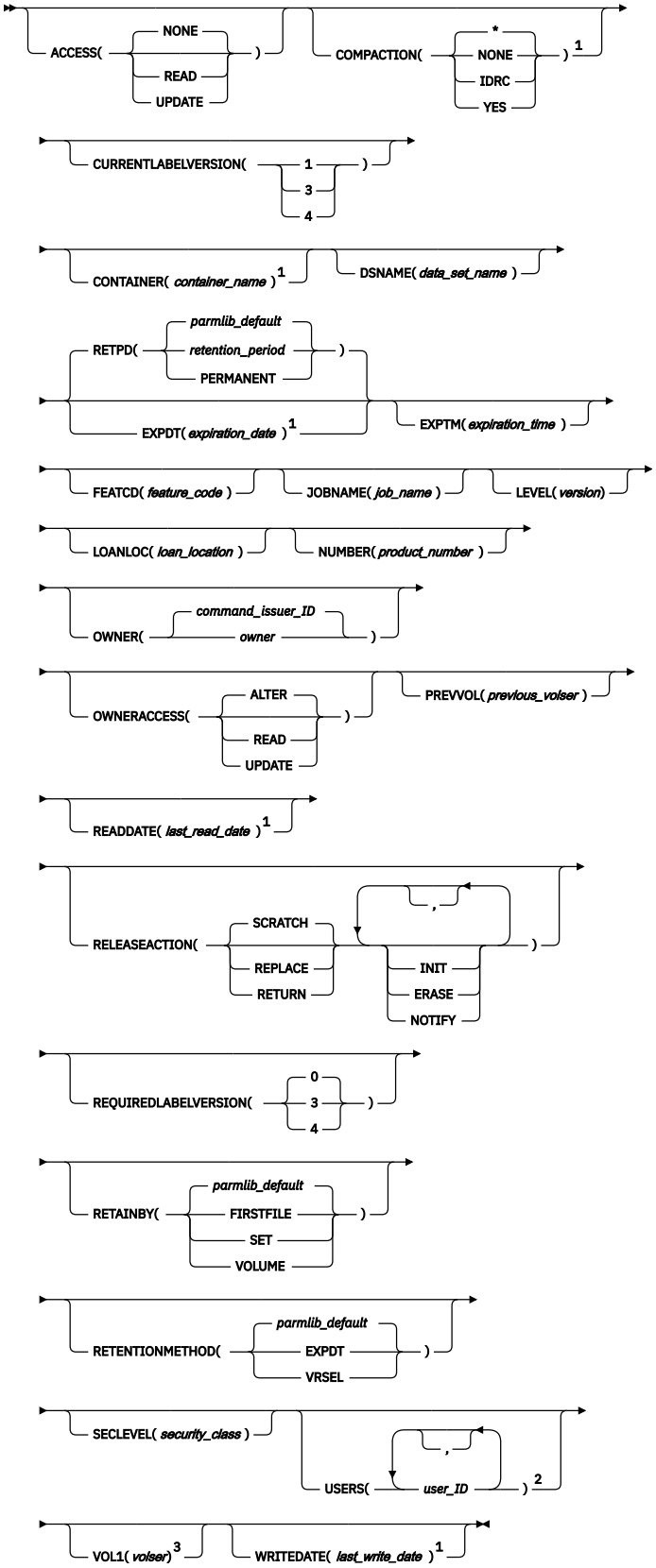


Notes:

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

Format of ADDVOLUME non-scratch volume optional operands

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.

Parameters of ADDVOLUME subcommand

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 these:

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.

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 2012/001. 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 12001. 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.

CAPACITY(*medinf | nn-mb*)
Use this operand to specify the volume capacity in megabytes (MB). DFSMSrmm normally sets the capacity of a volume based on the media type and the recording format, or you can manually set the capacity if the capacity cannot be determined using the media type and recording format. The MEDINF parmlib commands define the capacity of different combinations of media type and recording formats. There are built-in capacity values for IBM media types.

Specify a value between 0 and 4294967295.

COMPACTION(*|NONE|IDRC|YES)
Specifies the compaction technique used to record data on tape volumes. Use one of these:

*****
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 6 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 2012/001. 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 12001. 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.

CRDATE can be abbreviated as DATE.

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.

CRSYSID can be abbreviated as SYSID.

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.

CRTIME can be abbreviated as TIME.

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.

Notes:

1. When the DSNAME parameter is specified, a new data set record is created along with the volume record. This record is not assigned the default EXPDT, the default WHILECATALOG, nor the default LASTREFDAYS. Its expiration date is the same as that of the volume record.
2. DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these do not match to data sets with all uppercase characters.

DFSMSrmm does not check quoted data set names for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. Data set names without quotes must pass the following data set naming rules:

- A data set name can have one or more qualifiers.
- Each qualifier is one to eight characters. The first character alphabetic (A to Z) or national (# @ $). The remaining seven characters can be either alphabetic, numeric (0 - 9), national, or a hyphen (-).
- Qualifiers are separated by a period.
- DFSMSrmm adds your TSO PROFILE PREFIX value as the high-level qualifier.

- 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.

ERROR(READ(*count*),WRITE(*count*))
Use this operand to specify new error count values for read and write errors. These are assumed to be the permanent errors for the volume.

When you add a volume, the error counts are initially set to zero.

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 2012/001. 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 12001. 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.

When you specify the expiration date for a volume that is a part of a multivolume set managed by the EXPDT retention method retained by SET, DFSMSrmm updates the expiration date and time for all the volumes of the multivolume set. If you try to specify the expiration date for a volume managed by the EXPDT retention method retained by FIRSTFILE, the command will be rejected. The expiration date of the volume will be set to the expiration date of the first data set record on a single volume or on a volume set

When you specify the expiration date for a data set record representing one part of a multivolume data set on volumes managed by the EXPDT retention method, DFSMSrmm updates the expiration date and time for all the data set records for the data set. When you do not specify an expiration date, DFSMSrmm copies the expiration from the preceding data set record in the multivolume data set chain.

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 RETPD operand of the DFSMSrmm EDGRMM *xx* parmlib member.

EXPTM(*expiration_time*)
Specifies the time when the volume should be considered for release. The format for *expiration_time* is *hhmmss*, where:

hh
is hours

mm
is minutes

ss
is seconds

The expiration date is determined by the EXPDT or RETPD parameters. The time zone offset specified in the TZ options affects the resulting expiration time.

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 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.

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 characters other than blank, comma, and semicolon. You cannot use a generic jobname. Any jobname you use specific.

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

There is no default.

KEYENCODE1(H|L)
Specifies the encoding mechanism used for KEYLABEL1.

H
Public key hash

L
Label

When KEYLABEL1 is specified, the default value is L.

KEYENCODE2(H|L)
Specifies the encoding mechanism used for KEYLABEL2.

H
Public key hash

L
Label

When KEYLABEL2 is specified, the default value is L.

KEYLABEL1(*keylabel1_name*)
Specifies the key encryption key label number 1 for a non-scratch volume that is encrypted. A key label is 1-to-64 characters with blanks padding the field on the right. A key label contains alphanumeric, national, or special characters with some additional characters also allowed. Enclose it in single quotation marks if it contains any blanks or special characters.

When a volume is rewritten in a non-encryption format, DFSMSrmm does not clear the encryption key label related fields. Instead, the fields continue to be displayed until the volume is reused from scratch, or a release action causes them to be cleared.

When you specify this optional operand, you can also specify a value for the key encoding mechanism using the KEYENCODE1 operand.

KEYLABEL2(*keylabel2_name*)
Specifies the key encryption key label number 2 for a non-scratch volume that is encrypted. A key label is 1-to-64 characters with blanks padding the field on the right. A key label contains alphanumeric, national, or special characters with some additional characters also allowed. Enclose it in single quotation marks if it contains any blanks or special characters.

When a volume is rewritten in a non-encryption format, DFSMSrmm does not clear the encryption key label related fields. Instead, the fields continue to be displayed until the volume is reused from scratch, or a release action causes them to be cleared.

When you specify this optional operand, you can also specify a value for the key encoding mechanism using the KEYENCODE2 operand.

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 344 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 alphanumeric or national characters. 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, when NUMBER is specified. 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 these 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 can specify a distributed library name only if the library is an IBM Virtualization Engine.

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(*medianame*)
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 z/OS. 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 default MEDIANAME is the value you define with the EDGRMMxx parmlib OPTION MEDIANAME operand. For more information on how to set this value, see *z/OS DFSMSrmm Implementation and Customization Guide*

MEDIATYPE(* | CST | ECCST | EHPCT | HPCT | MEDIA5 | MEDIA6 | MEDIA7 | MEDIA8 | MEDIA9 | MEDIA10 | MEDIA11 | MEDIA12 | MEDIA13 | *medinf_mediatype*)
Specifies the volume's physical media type. Use one of these:

*****
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 Enterprise Tape Cartridge

MEDIA6/EWTC
IBM Enterprise WORM Tape Cartridge 3592

MEDIA7/EETC
IBM Enterprise Economy Tape Cartridge 3592

MEDIA8/EEWTC
IBM Enterprise Economy WORM Tape Cartridge 3592

MEDIA9/EXTC
IBM Enterprise Extended Tape Cartridge 3592

MEDIA10/EXWTC
IBM Enterprise Extended WORM Tape Cartridge 3592

MEDIA11/EATC
IBM Enterprise Advanced Tape Cartridge

MEDIA12/EAWTC
IBM Enterprise Advanced WORM Tape Cartridge

MEDIA13/EAETC
IBM Enterprise Advanced Economy Tape Cartridge

medinf_mediatype
Specify a non-IBM media type if your installation definition contains media information for *medinf_mediatype* that matches the media information assigned to the volume.

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 28 for information about how DFSMSrmm assigns MEDIATYPE and RECORDINGFORMAT.

The default value is *.

MEDINF(*medinf_name*)
Specifies the assigned installation-defined media information to the volume. This value is one-to-eight alphanumeric characters and defined once in your installation. You can use the LISTCONTROL subcommand with the MEDINF operand to display the media information. For more information about MEDINF, see *z/OS DFSMSrmm Implementation and Customization Guide*.

Default: IBM.

NOWORM
Use this operand to identify that the volume is not a WORM volume. You cannot specify this operand for a volume recorded automatically by DFSMSrmm during open processing.

Use the WORM operand to set the WORM attribute.

NOWORM is the default value.

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.

OPENCOUNT(*count*)
Use this operand to specify how many times any data set on the volume has been opened.

When you add a volume, the open count is initially set to zero.

The value range is 0 to 65535.

ORIGINALEXPDT(*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 2012/001. 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 12001. 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.

`ORIGINALEXPDT` can be abbreviated as `OEXPDT`.

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.

When the RACF TAPEVOL class is active, and TPRACF(P) or TPRACF(A) is in effect, DFSMSrmm uses the OWNERACCESS information to build the RACF TAPEVOL access list. OWNERACCESS can be used together with OWNER to define the initial RACF TAPEVOL volume profile access, specifying the type of access the volume owner has to a volume.

The OWNERACCESS value can be one of these:

ALTER
The volume owner is allowed to read from the tape volume, to write add and delete data sets to the volume, and to create or destroy tape volume labels through OPEN or end-of-volume operations. For discrete tape volume profiles, the volume owner is allowed to change the profile, including the access list.

ALTER is the default value.

READ
The volume owner has only read access.

UPDATE
The volume owner is allowed to read from the tape volume, and to write additional data sets to the volume.

This operand is ignored for scratch volumes.

For more information, refer to the topic Maintaining the user access list in *z/OS DFSMSrmm Implementation and Customization Guide*.

PERCENT(*0 | percent*)
Use this operand to specify how full the volume is. You can specify a value between 0 and 100.

When you add a volume, the percent full is initially set to zero.

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 327 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 28 on page 212. 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.

PREVVOL is mutually exclusive with RETENTIONMETHOD and RETAINBY.

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 previously defined and empty.

If you do not supply a pool prefix or a rack number, DFSMSrmm assigns the volume a rack number as described in Table 28 on page 212. 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 2012/001. 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 12001. 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 | EFMT2 | EEFMT2 | EFMT3 | EEFMT3 | EFMT4 | EEFMT4 | *medinf_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 Enterprise Format 1 recording technology. You can only specify EFMT1 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), and MEDIATYPE(MEDIA8).

EFMT2
Data has been written to the volume in Enterprise Format 2 recording technology. You can only specify EFMT2 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), MEDIATYPE(MEDIA8), MEDIATYPE(MEDIA9) and MEDIATYPE(MEDIA10).

EEFMT2
Data has been written to the volume in Enterprise Encrypted Format 2 recording technology. You can only specify EEFMT2 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), MEDIATYPE(MEDIA8), MEDIATYPE(MEDIA9) and MEDIATYPE(MEDIA10).

EFMT3
Data has been written to the volume in EFMT3 (enterprise format 3) recording format. A recording format of EFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EEFMT3
Data has been written to the volume in EEFMT3 (enterprise encrypted format 3) recording format. A recording format of EEFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EFMT4
Data has been written to the volume in EFMT4 (enterprise format 4) recording format. A recording format of EFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) only.

EEFMT4
Data has been written to the volume in EEFMT4 (enterprise encrypted format 4) recording format. A recording format of EEFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) only.

medinf_recordingformat
Specify a non-IBM media recording format if your installation definition contains media information for *medinf_recordingformat* that matches the media information assigned to the volume.

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 28 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 these:

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 these 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 these 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.

RETAINBY(FIRSTFILE | SET | VOLUME)

Specifies how DFSMSrmm is to retain an EXPDT-retained volume or multivolume set:

FIRSTFILE
The expiration date of the first file is used to set the expiration date of a single volume or a multivolume set. All volumes in a multivolume set will have exactly the same expiration date and will be released to scratch in the same run of DFSMSrmm inventory management.

Additional data sets added later to a volume or to a multivolume set can have different expiration dates that are independent of the volume expiration date.

SET
DFSMSrmm uses the highest expiration date of all volumes in the set and each file on a volume set can increment it. All volumes in the set will have exactly the same expiration date and will be released to scratch on the same run of DFSMSrmm inventory management.

VOLUME
The expiration date of the volume is considered for each volume separately and each file on a volume can increment the volume expiration date.

Specify this operand for the first volume in a multivolume sequence. All other volumes added to the set will assume the same RETAINBY value.

Note:

1. The RETAINBY operand cannot be specified for a volume managed by the VRSEL retention method. Use the RETAINBY operand only for volume sets that use the EXPDT retention method.
2. When a RETAINBY value is defined for a non-scratch volume, it is not overridden to the default during OPEN output processing, but can be changed using CHANGEVOLUME subcommand.
3. The RETAINBY operand cannot be specified if the PREVVOL operand is specified.
4. The RETAINBY operand is ignored if you use STATUS(SCRATCH) or STATUS(VOLCAT).

Default: If RETAINBY is omitted, the RETENTIONMETHOD(EXPDT(RETAINBY(value)) specified in parmlib is used.

RETENTIONMETHOD(EXPDT | VRSEL)
Use this operand to set the retention method for a tape volume set. Specify this operand for the first volume in a multivolume sequence. All other volumes added to the set assume the same retention method.

Once a retention method is defined for a non-scratch volume, it is not overridden to the system-wide default during OPEN output processing, but can be changed by installation exit EDG_EXIT100. Volumes in a set always assume the retention method of the first volume in the set.

Specify EXPDT to set the retention method for a tape volume set to be based on EXPDT. Data sets and volumes managed by this retention method are never processed by VRSEL inventory management. If the retention method is changed from VRSEL to EXPDT, then the RETAINBY field is set from the AV command, if specified, or from the parmlib. The expiration date of the volume or multivolume set is updated according to the RETAINBY value.

Specify VRSEL to set the retention method for a tape volume set to be VRSEL. This option enables DFSMSrmm inventory management to attempt to match data sets and volumes to vital record specifications, and if a match is found, to determine if the data set or volumes are to retained by VRS.

See Chapter 3, “Retention methods,” on page 45 for considerations for selecting a retention method.

RETENTIONMETHOD is mutually exclusive with PREVVOL. RETENTIONMETHOD is ignored if used with STATUS(SCRATCH) or STATUS(VOLCAT).

RETENTIONMETHOD can be abbreviated as RM.

Default: If RETENTIONMETHOD is omitted, the retention method specified in parmlib is used.

RETPD(*retention_period*)
Specifies the number of days that DFSMSrmm retains the volume before considering it for release. *retention_period* is a decimal number from 0 to 93000, or PERMANENT. The retention period is added to the current date and time to compute the EXPDT and the EXPTM. RETPD(0) causes the default retention period to be used for the volume as defined in the PARMLIB.RETPD cannot exceed the maximum retention period (multivolume) set by your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

When you specify the retention period for a volume that is a part of a multivolume set managed by the EXPDT retention method and retained by SET, DFSMSrmm updates the expiration date for all the volumes of the multivolume set.

When you specify the retention period for a volume managed by the EXPDT retention method and retained by FIRSTFILE, the specified retention period will be ignored. The retention period of the volume will be set to the retention period of the first data set record on a single volume or on a volume set.

When you do not specify a retention period, DFSMSrmm copies the retention period from the preceding data set record in the multivolume data set chain.

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 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 327 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
STATUS(SCRATCH|MASTER|USER|VOLCAT) specifies the volume's status. Use one of these:

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
STORAGEGROUP(*storage_group_name*) specifies the SMS-defined storage group to which the volume belongs. A storage group name is one-to-eight characters other than blank, comma, and semicolon. 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. STORAGEGROUP can be abbreviated as STORGRP.

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.

For system-managed manual tape library volumes, the storage group name is used for scratch pool validation only when you request that a specific storage group name is used for pooling. For all non-system managed scratch pooling validation, the storage group name is significant and is always used to ensure that a volume from the correct pool is mounted.

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. 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.

TYPE(PHYSICAL) can be used for volumes that do not reside in a VTS. Non system- managed virtual tape libraries can use either TYPE(PHYSICAL) or TYPE(LOGICAL). However, TYPE(LOGICAL) and TYPE(STACKED) cannot be specified for volumes in a system-managed tape library that is not a VTS. which includes a non-IBM VTS that is being managed as a manual tape library.

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

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.

- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

USE(IRMM,MVS,VM)
Specifies the operating systems where the volume can be used. You can select one or more of IRMM, MVS, and VM. To indicate multiple operating systems are valid, enter the values 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. Vendor information enclosed in single quotation marks if it contains any special characters or blanks. 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.

WMC(*count*)
Use this operand to set the write mount count for any volume. The write mount count reflects how many times the volume has been mounted for output and actually written to while mounted.

When you add a volume, the write mount count is initially set to zero.

For WORM volumes mounted and processed under DFSMSrmm control on z/OS, the WMC value is obtained from the tape drive and can then no longer be changed by commands.

The value range is 0 to 65535.

WORLDWIDEID(*world_wide_identifier*)
Use this operand to specify the unique world wide identifier set on the volume by the manufacturer. The world wide ID is 12 characters in hexadecimal. You can specify the value as 12 characters, a hex string of 24 characters (x'...'), or as a binary string (b'....'). When entered in hex, you specify 24 characters 0-9, A-F. An example of a world wide identifier is WWID(x’12345678ABCDEF090000FFEE’). When you specify a character string, it can be any string of 12 alphanumeric, national, special, or EBCDIC text characters, enclosed in quotes when special or EBCDIC text characters are specified. DFSMSrmm converts your character string to hex.

WORLDWIDEID can be abbreviated as WWID.

The WWID is maintained and displayed as a hexadecimal value and is displayed using 24 characters.

In some publications, the world-wide unique cartridge identifier (WWCID) can also be referred to as the world-wide identifier (WORLDWIDEID or WWID).

There is no default.

Do not use the DFSMSrmm subcommands to set or change the WWID value. Instead, add volumes as you normally would do, and DFSMSrmm records the value when the volume is first used. Subsequent use of the volume causes DFSMSrmm to ensure that the recorded WWID and the WWID obtained from the mounted volume both match. If the values do not match, the volume is rejected. Once the WWID is set by command or recorded by DFSMSrmm when the volume was used while mounted, you cannot change the value in the DFSMSrmm control data set. If the WWID is incorrect in DFSMSrmm, your choice is to either delete and then re-add the volume, or to use RMM REPLACE processing. If DFSMSrmm already has data set details for the volume, keep a record of these so they can be added back by using the DFSMSrmm subcommands.

WORM
Use this operand to identify that the volume is a WORM volume. When you set the WORM attribute and the volume is OPENed for input or output on z/OS, DFSMSrmm volume validation checks that the write mount count and worldwide unique volume ID values recorded in DFSMSrmm match those obtained from the cartridge and maintained by the drive.

You cannot specify this operand for a WORM volume recorded automatically by DFSMSrmm during open processing.

NOWORM is the default value.

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 2007/001. 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 12001. 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.

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 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 z/OS 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 this 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') WORM
```

Return codes of ADDVOLUME subcommand

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose of ADDVRS subcommand

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 policies for volumes managed by the VRSEL retention method and the data sets on those volume that are not excluded from VRSEL processing.

When a vital record specification is added, no data set or volume information is changed. During the next vital records processing run, DFSMSrmm uses the defined vital record specifications, both the old ones and the newly added one, to apply policies. If the data set or volume now matches to another 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.

When you add, change, or delete a vital record specification, you should run VRSEL with the VERIFY option, as described in "Maintaining your vital record specifications" on page 109.

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 5, "Using vital record specifications to retain and move volumes," on page 73 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. 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 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*

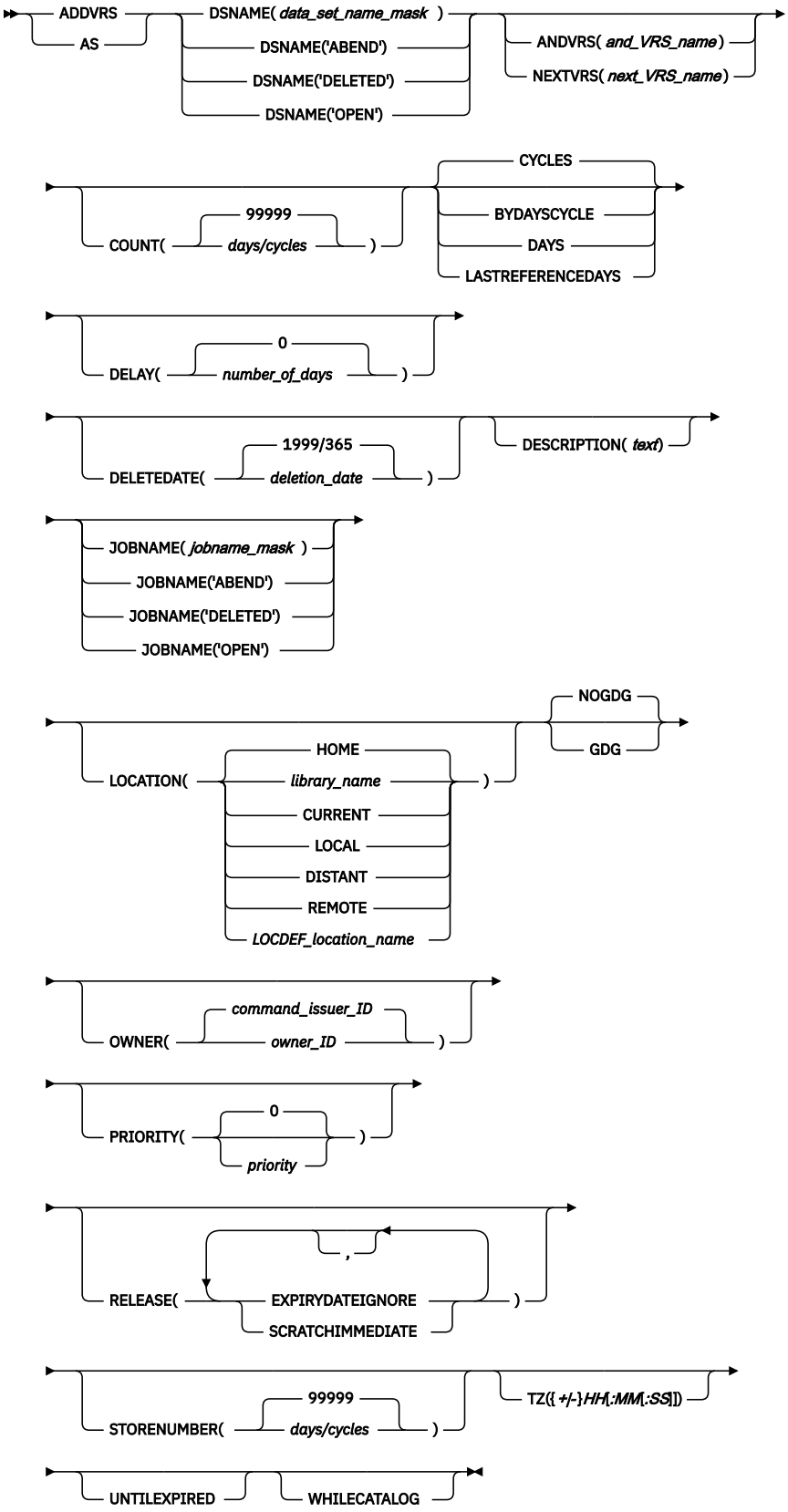
Guide for more information on assigning management class names and vital record specification management values.

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. 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.

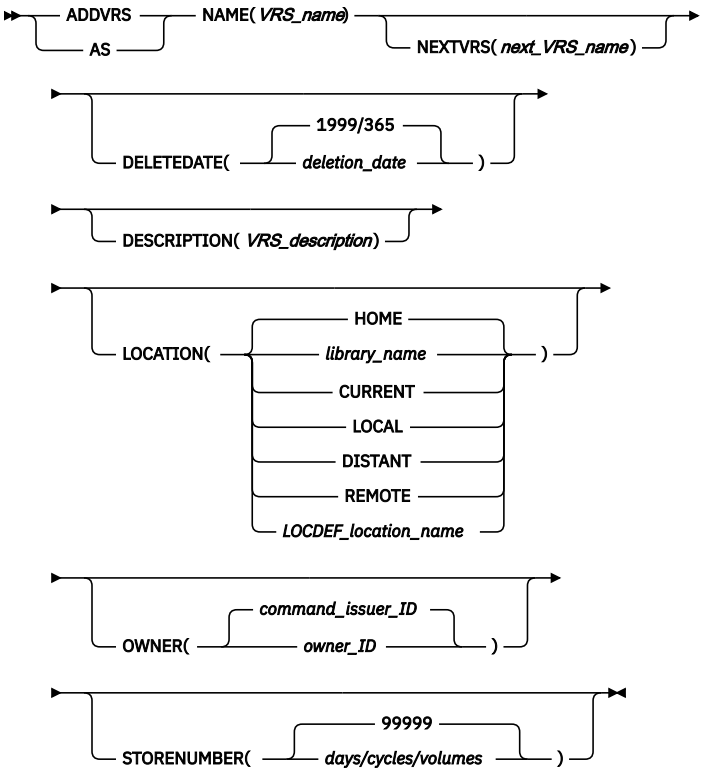
Format of ADDVRS DATASET syntax

ADDVRS DATASET Syntax



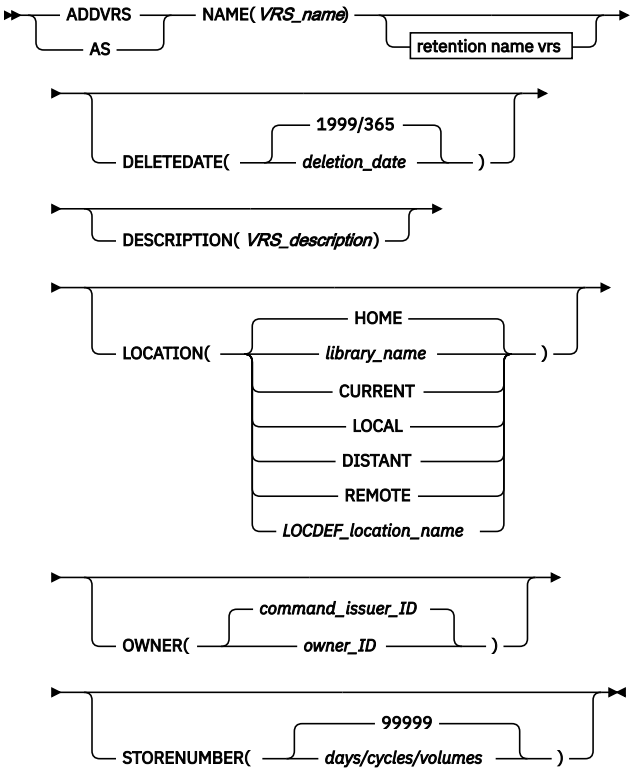
Format of ADDVRS location NAME syntax

ADDVRS Location NAME Syntax

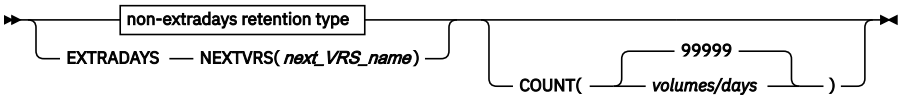


Format of ADDVRS retention NAME syntax

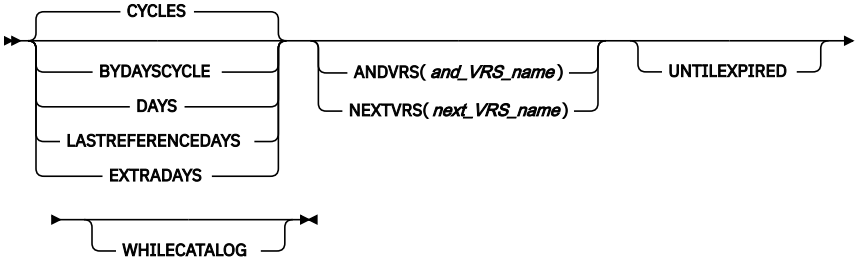
ADDVRS Retention NAME Syntax



retention name vrs

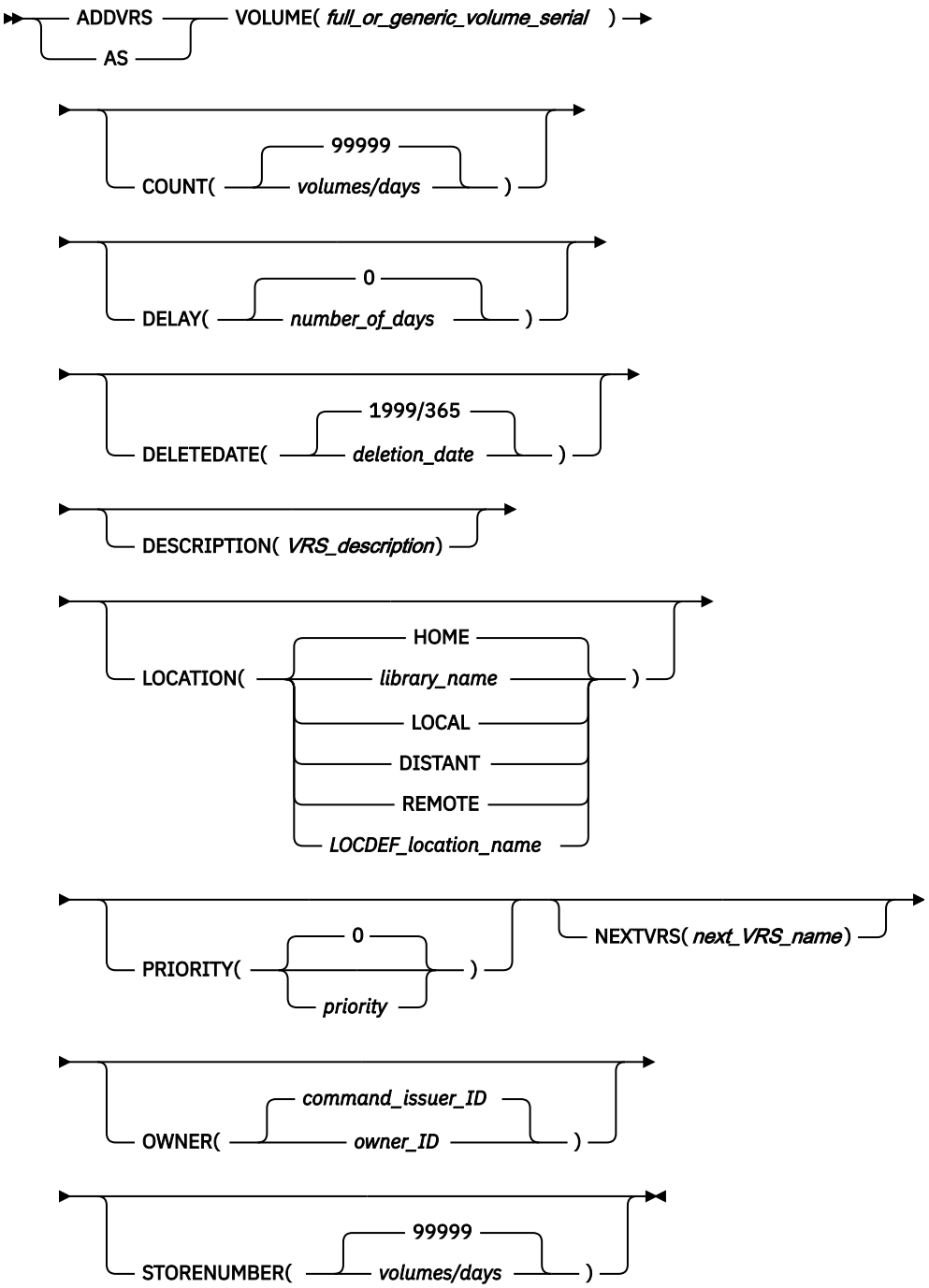


non-extradays retention type



Format of ADDVRS VOLUME syntax

ADDVRS VOLUME Syntax



Parameters of ADDVRS subcommand

ANDVRS(*and_VRS_name*)
Specifies creating a chain of vital record specifications. The chain contains all the retention conditions that 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.

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. (For more

information on cycle retention, see the description of the GDG parmlib option in *z/OS DFSMSrmm Implementation and Customization Guide*.)

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 for volume vital record specifications. 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.

The value range for data set name vital record specifications and retention NAME vital record specifications is 0 to 99999.

When you use the DELAY operand with the DAYS and LASTREFERENCEDAYS operands, the COUNT value you specify greater than or equal to the total of the DELAY value and the STORENUMBER value. For the CYCLES and BYDAYSCYCLE operands, the COUNT value greater than or equal to the STORENUMBER value.

For a NAME VRS, the COUNT range is 0 to 99999. There is no default.

DFSMSrmm validates the COUNT value as follows:

- COUNT 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 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 more information on GDG cycles, see the description of the GDG parmlib option in *z/OS DFSMSrmm Implementation and Customization Guide*.) 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 11:59 PM on November 12th after running vital record processing on the 12th. If you run vital record processing at 12:01

AM 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 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 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 2012/001. 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 12001. 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.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

DFSMSrmm does not check quoted data set names for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. Also, data set mask naming rules met. Data set names without quotations must pass the data set and data set mask naming rules.

Data set names without quotations and data set name masks without quotations must pass these data set naming rules:

- A data set name can have one or more qualifiers.
- Each qualifier is one to eight characters. The first character alphabetic (A to Z) or national (# @ $). The remaining seven characters can be either alphabetic, numeric (0 - 9), national, or a hyphen (-).
- Qualifiers are separated by a period.
- DFSMSrmm adds your TSO PROFILE PREFIX value as the high-level qualifier.
- The data set name must not include a member name.

In addition, data set name masks must pass these data set mask naming rules:

- 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 67 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.

Period (.)
A leading or trailing period is not allowed. Consecutive periods are also not allowed.

Double asterisk (**)
Cannot be specified within a data set name qualifier.

Cannot be specified before a qualifier with a single asterisk. For example, specifying DSNAME(**.*) is not a valid data set name mask.

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 z/OS 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 DELETED to manage all data sets that have the 'deleted' flag set on. 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.

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 these 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. (For more information on GDG cycle retention, see the description of the GDG parmlib option in *z/OS DFSMSrmm Implementation and Customization Guide*.)

JOBNAME(*jobname_mask*)
Specifies a job name that created a data set. A job name is one-to-eight alphanumeric characters or $, #, 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.

You can specify the job 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. Use the DSNAME operand to specify a data set name mask.

You can specify the job name mask DELETED to manage all data sets that have the 'deleted' flag set on. Use the DSNAME operand to specify a data set name mask.

You can specify the job 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. Use the DSNAME operand to specify a data set name mask.

If you have data sets with job names that include symbols other than alphanumeric characters, $, #, or @, use a job name mask to cover them.

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 CURRENT if you want to retain a volume in its current location, instead of moving it to a storage location or the home location.

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 211 and "CHANGEVOLUME: Changing volume information" on page 269 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 can specify a distributed library name only if the library is an IBM Virtualization Engine.

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. 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 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.

DFSMSrmm's handling of the vital record specification release options enables them to be applied to a volume from data set name vital record specifications even when the data set is not retained by a vital record specification. Release options are applied for any data set that matches to a vital record specification. This means that you can return a data set to scratch on the same day that it is created even when it is never retained by a data set name vital record specification.

When there are multiple data sets on a volume, the results for release option processing are such that:

- If any data set on a volume is or has ever been retained by a vital record specification, the release options for the volume are set only from data sets that are retained by the vital record specification.
- The handling of the release options as data sets are dropped from a vital record specification retention is unchanged.
- If no data sets on a volume are vital record specification retained and none of them have yet been retained by a vital record specification, and the volume is not yet retained by a volume vital record specification, the release options are taken from any data sets that match to a vital record specification. Both primary and secondary vital record specification matches are considered.
- In the case there is a data set(s) on the volume that matches a vital specification along with one or more data sets on the volume that are not matched to vital record specifications, the release options of the matched data set will be honored. This could lead to premature expiration of the non matched data set. The recommendation is that if Vital record specification(VRS) expiration method is being used; all data sets should have a VRS policy defined. Normally a default VRS policy for dsn ** will prevent this issue and DFSMSrmm will assign the highest retention date from all data sets managed by VRS for the volume or volume set.

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.

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 0-99999, where 99999 indicates that all remaining data sets or volumes are to be retained. The default is STORENUMBER(99999).

STORENUMBER less than or equal to COUNT.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

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 both a primary vital record specification and secondary vital record specification, and UNTILEXPIRED is specified on the primary vital record specification, DFSMSrmm uses the retention information from the secondary vital record specification, rather than the volume expiration date, to determine if the data set should be retained. UNTILEXPIRED allows primary vital record specifications to manage location, without retaining data sets longer that the secondary vital record specification retains it.

The UNTILEXPIRED retention uses the volume expiration date when used in a management value or management class (secondary) vital record specification.

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.

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 these 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 moves 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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

Before you begin:

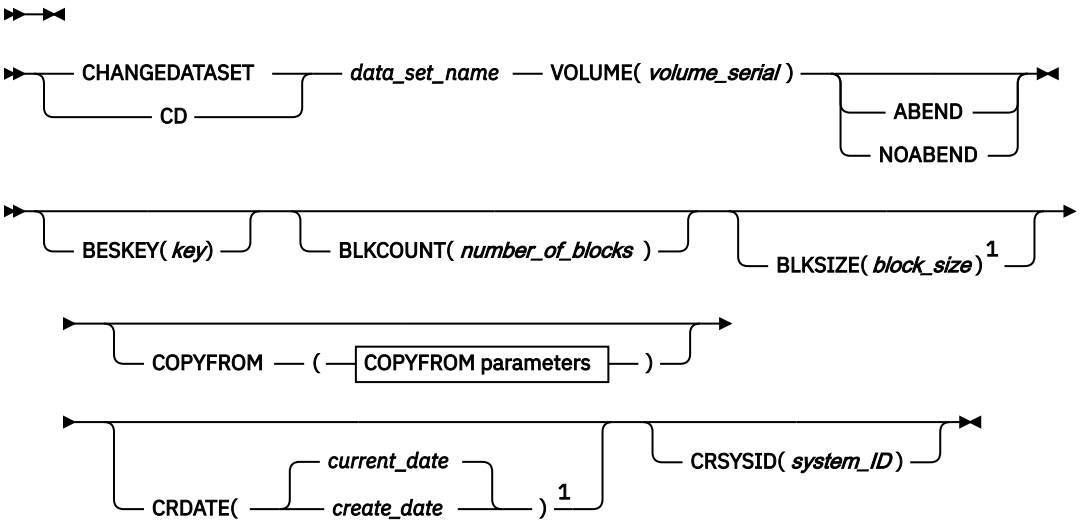
- To use the RMM CHANGEDATASET subcommand, you need UPDATE access to the STGADMIN.EDG.MASTER resource profile to change information in non-restricted fields for your own volumes.
- To use the RMM CHANGEDATASET FORCE subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile and UPDATE 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.

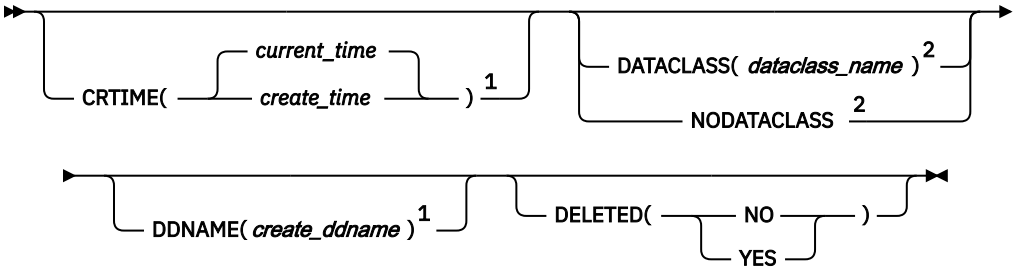
Related Reading: See *z/OS DFSMSrmm Implementation and Customization Guide* for information about using the DFSMSrmm parmlib OPTION COMMANDAUTH command and authorizing the use of the DFSMSrmm subcommands.

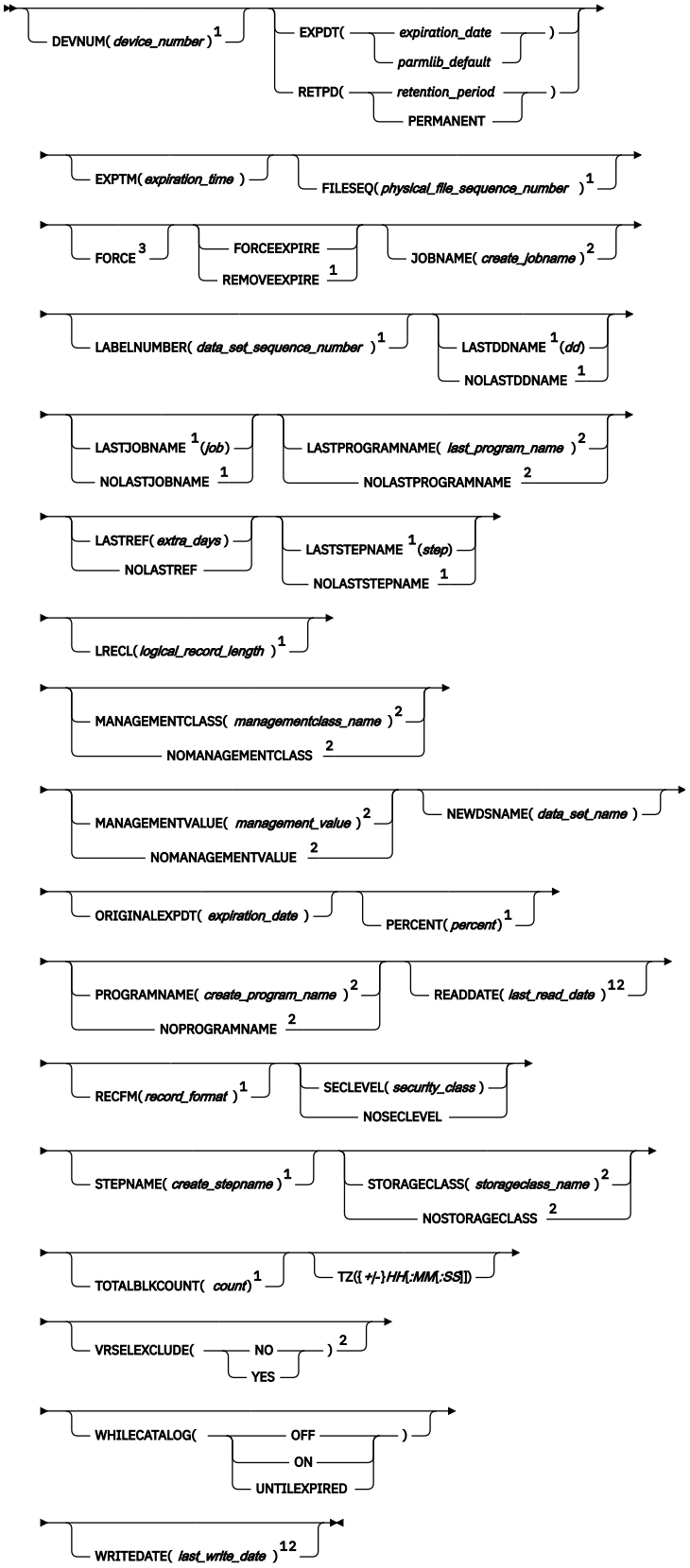
Use the CHANGEDATASET subcommand to update information about data sets defined to DFSMSrmm. Give a data set name and the serial number of the volume where it resides and, optionally, the data set's relative position on the volume. DFSMSrmm assumes that the data set is the first one on the specified volume 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 337 for more information.

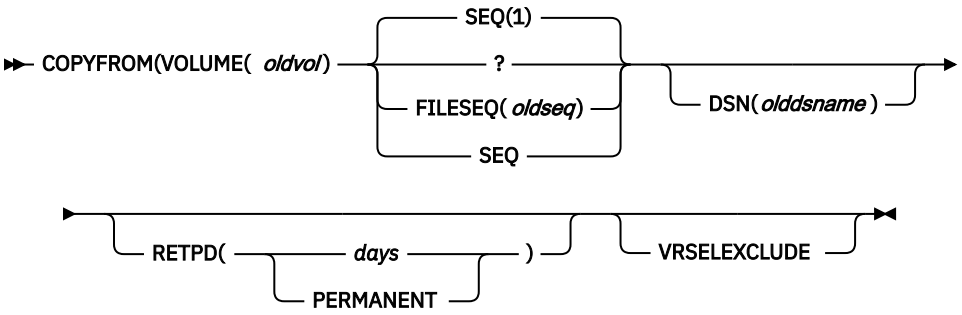
Format




CRTIME(
current_time
create_time
) 1
DATACLASS(dataclass_name) 2
NODATACLASS 2
DDNAME(create_ddname) 1
DELETED(
NO
YES
)



COPYFROM parameters



Notes:

[1] When this operand is specified, the FORCE operand must also be specified if DFSMSrmm has automatically recorded information about the volume during O/C/EOV processing of a data set on the volume. 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.

[2] This operand can be specified only 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.

Parameters

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.

BESKEY(*key*)
Use the BESKEY operand to set or change the CA Tape Encryption key index, which is set by the BES subsystem. *key* is an encryption key index and can be expressed as a number from 0 to 2147483647, or a hexadecimal value from X'00' to X'FFFFFFFF'.

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 label. The minimum allowable decimal value is 0; the maximum allowable decimal value is 4294967295. DFSMSrmm uses BLKCOUNT together with BLKSIZE to calculate the data set's approximate size and the sum of all data set sizes to set the volume usage.

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 and the sum of all data set sizes to set the volume usage. If the volume is OCE recorded, you can only specify this operand if you are authorized to use the FORCE operand.

COPYFROM(*suboperands*)
Specifies a source data set whose attributes are to be copied when creating the meta data for the target data set. DFSMSrmm copies all attributes that are not related to the physical aspects of the data set, volume, and tape drive.

These suboperands can be used:

DSN(olddsname)
Identifies the source data set record from which attributes are to be copied. You can optionally use a different data set name as the target. The default is that the olddsname matches data_set_name

VOLUME(oldvol)
Identifies the source data set record from which attributes are to be copied. There is no default value.

FILESEQ(oldseq)
Identifies the source data set record from which attributes are to be copied. It specifies the physical file sequence number. Use this operand to identify the relative position of the source data set on the oldvol. The minimum allowable decimal value is 1. The maximum allowable decimal value is 65535. The default value is 1.

RETPD(daysUntilFromDatasetIsNoLongerRetained|PERMANENT)
Causes DFSMSrmm to update the source data set record expiration date. There is no default value. By default, the source data set is not updated. The value can be 0 to 93000. If PERMANENT is specified, the expiration date is set to 1999/365, which means that the data set will never expire.

VRSELEXCLUDE
Causes the source data set to be excluded from VRSEL processing. VX is an alias of VRSELEXCLUDE.

When you specify any other CHANGEDATASET subcommand operands, DFSMSrmm will process the COPYFROM operand first, then the additional operands. This means that additional operands can specify data that overrides the attributes copied.

CRDATE(*create_date*|*current_date*)
Specifies the date when the data set was written to tape. CRDATE can be abbreviated as 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 2012/001. 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 12001. 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(*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. 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.

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. You can specify CRSYSID if you are authorized to use the FORCE operand.

CRSYSID can be abbreviated as **SYSID**.

CRTIME(*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.

CRTIME can be abbreviated as TIME.

DATACLASS(*dataclass_name*)
Specifies the data class name that created the data set. *dataclass_name* is one-to-eight characters other than blank, comma, and semicolon. 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 no default value.

data_set_name
Specifies the name of the data set being changed. The name is 1 to 44 characters in length and enclosed in quotes if any special characters are included. If the data set name is not enclosed in quotes, your TSO PROFILE PREFIX value is applied.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

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 characters other than blank, comma, and semicolon. You cannot use a generic DD name. You can specify this operand if you are authorized to use the FORCE operand.

DELETED(NO | YES)
Specifies the data set is deleted.

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 multivolume 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 2012/001. 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 12001. 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.

When you change the expiration date for a data set record representing one part of a multivolume data set on volumes managed by the EXPDT retention method and retained by VOLUME or SET, DFSMSrmm updates the expiration date and time for all the records of the data set.

When you change the expiration date of a first data set record on a single volume, or on a volume set, managed by the EXPDT retention method and retained by FIRSTFILE, DFSMSrmm updates the expiration date and time of the volume, or all the volumes of the multivolume set, with this value. When you change the expiration date of any other data set record of the volume or volume set retained by FIRSTFILE, the data set record is updated, but will have no effect on the expiration date of the volume or volume set.

EXPDT is mutually exclusive with RETPD. Changing the expiration date does not change the expiration time of the data set.

EXPTM(*expiration_time*)
Specifies the time when the data set should be considered for release. The format for *expiration_time* is *hhmmss*, where:

hh
is hours

mm
is minutes

ss
is seconds

FILESEQ(*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. FILESEQ can be abbreviated as SEQ.

FORCE
Overrides 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.

FORCEEXPIRE
Use FORCEEXPIRE to block access to a data set and uncatalog it. The result of the uncatalog call depends on the setting of the UNCATALOG RMM PARMLIB option.

UNCATALOG(NO)
Prevents FORCEEXPIRE from uncataloging the data set.

UNCATALOG(YES)
Makes sure FORCEEXPIRE uncatalogs the data set.

UNCATALOG(SCRATCH)
Only uncatalogs data sets on scratch volumes.

FORCEEXPIRE causes "(ForceExpire)" to appear next to the expiration date in the RMM LISTDATASET subcommand output.

All attempts to read or write to the data set fail with ABEND S413.

For NL (not labeled) tapes, a data set that has the FORCEEXPIRE bit set is blocked even if the JCL has the wrong data set name, as long as the sequence number of the data set matches that stored in the RMM CDS. BLP (bypass label processing) cannot be used to read or write data sets that have the FORCEEXPIRE bit set.

FORCEEXPIRE does not change the actual expiration date of the data set or the volume on which it resides. FORCEEXPIRE can affect the retention of the data set when WHILECATALOG is used.

The RMM CHANGEDATASET subcommand specified with the REMOVEEXPIRE and FORCE operands is the only way to regain access to the data set after a FORCEEXPIRE. You can specify the REMOVEEXPIRE operand if you are authorized to use the FORCE operand.

JOBNAME(*create_jobname*)
Specifies the job name that created the data set. A job name is one-to-eight characters other than blank, comma, and semicolon. You cannot use a generic job name. Any job name you use 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.

LASTDDNAME(*dd*)
Use this operand to specify the name of the DD statement that last referenced the data set. A DD name is one-to-eight characters other than blank, comma, and semicolon. You cannot use a generic DD name.

LASTJOBNAME(*job*)
Use this operand to specify the name of the job that last referenced the data set. A job name is one-to-eight characters other than blank, comma, and semicolon. You cannot use a generic jobname. Any jobname you use specific.

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 characters other than blank, comma, and semicolon. You can specify this operand if you have CONTROL access to STGADMIN.EDG.MASTER security resource.

LASTREF(*extra_days*)
LASTREF(extra_days) specifies the number of days that the data set will be retained after the data set was last referenced.

LASTREF(*extra_days*) applies only to data sets on volumes managed by the EXPDT retention method. LASTREF cannot be specified for data sets on volumes managed by the VRSEL retention method.

extra_days is a decimal number between 0 and 93000. The value must not exceed the maximum retention period multivolume specified in the DFSMSrmm EDGRMMxx parmlib member. An *extra_days* value of 0 has the same effect as using NOLASTREF. DFSMSrmm uses the LASTREF extra days to evaluate the data set expiration date. The extra days are added to the date of last reference. The data set expiration date is set to the maximum of the date calculated using LASTREF extra days, and the current expiration date. Any reference to the data set by a read or write operation will redetermine the expiration date.

When a file is added to a multivolume data set, the LASTREF or NOLASTREF attribute is copied from the preceding file. For a volume set retained by VOLUME or SET DFSMSrmm ensures that the LASTREF(*extra_days*) or NOLASTREF data set attribute is the same for all files of a multivolume data set. For a volume set retained by FIRSTFILE no additional processing is performed to keep the LASTREF extra days attribute consistent across the multivolume data set, because the expiration date depends only on the LASTREF extra days attribute of the first file of the first volume.

LASTSTEPNAME(*step*)
Use this operand to specify the name of the step that last referenced the data set. A step name is one-to-eight characters other than blank, comma, and semicolon. You cannot use a generic step name.

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 that created the data set. A management class name is one-to-eight characters other than blank, comma, and semicolon. 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.

For a data set on a volume that is managed by the VRSEL retention method, the new management class name can be used for matching vital record specifications during VRSEL processing. However, no attributes from the new management class are used for an existing data set, regardless of the retention method used.

There is no default value.

MANAGEMENTVALUE(*VRS_management_value*)
Specifies the vital record specification management value that created the data set. A vital record specification management value is one-to-eight characters other than blank, comma, and semicolon. 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.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

DFSMSrmm does not check quoted data set names for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. Data set names without quotes must pass these data set naming rules:

- A data set name can have one or more qualifiers.
- Each qualifier is one to eight characters. The first character alphabetic (A to Z) or national (# @ $). The remaining seven characters can be either alphabetic, numeric (0 - 9), national, or a hyphen (-).
- Qualifiers are separated by a period.
- DFSMSrmm adds your TSO PROFILE PREFIX value as the high-level qualifier.
- The data set name must not include a member name.

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.

NODATACLASS
Specifies that you want the DATACLASS value set to blanks.

NOLASTDDNAME
Use this operand to ensure there is no name of a DD statement that last referenced the tape data set.

NOLASTJOBNAME
Use this operand to ensure there is no name of a job that last referenced the tape data set.

NOLASTPROGRAMNAME
Specifies the NOLASTPROGRAMNAME condition. Specify this operand to remove the name of the program that last referenced the data set.

NOLASTREF
NOLASTREF specifies that DFSMSrmm does not consider the data set last reference date when evaluating the data set expiration date.

NOLASTREF applies only to data sets on volumes managed by the EXPDT retention method. NOLASTREF cannot be specified for data sets volumes managed by the VRSEL retention method.

When a file is added to a multivolume data set, the LASTREF or NOLASTREF attribute is copied from the preceding file. For a volume set retained by VOLUME or SET DFSMSrmm ensures that the LASTREF(*extra_days*) or NOLASTREF data set attribute is the same for all files of a multivolume data set. For a volume set retained by FIRSTFILE no additional processing is performed to keep the LASTREF extra days attribute consistent across the multivolume data set, because the expiration date depends only on the LASTREF extra days attribute of the first file of the first volume.

NOLASTSTEPNAME
Use this operand to ensure there is no name of a step that last referenced the tape data set.

NOMANAGEMENTCLASS
Specifies that you want the MANAGEMENTCLASS value set to blanks.

NOMANAGEMENTVALUE
Specifies that you want the MANAGEMENTVALUE value set to blanks.

NOPROGRAMNAME
Specifies the NOPROGRAMNAME condition. Specify to remove the name of the program that created the data set.

NOSECLEVEL
Specify to remove the security classification from a data set.

NOSTORAGECLASS
Specifies that you want the STORAGECLASS value set to blanks.

ORIGINALEXPDT(*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 2012/001. 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 12001. 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.

`ORIGINALEXPDT` can be abbreviated as `OEXPDT`.

PERCENT(*nnn*)
Specifies the percentage of the volume occupied by the data set. The percentage value (*nnn*) range is 0 to 100.

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 characters other than blank, comma, and semicolon. 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 2012/001. 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 12001. 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 you want to set a date in the future, the FORCE operand is required.

If a vital record specification or a data set LASTREF attribute indicates that DFSMSrmm retains 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. 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 ISO/ANSI record

DB
Blocked variable-length ISO/ANSI record

DS
Variable-length ISO/ANSI spanned record

DBS
Variable-length ISO/ANSI blocked spanned record

You can also append either A or M to the fixed and variable formats.

A
The record contains ISO/ANSI 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.

REMOVEEXPIRE
Resets the FORCEEXPIRE flag for the data set, allowing it to be accessed. You must specify the FORCE operand with REMOVEEXPIRE.

RETPD(*retention_period*)
Specifies the number of days that DFSMSrmm retains the data set before considering it for release. *retention_period* is a decimal number from 0 to 93000. The retention period is added to the current date and time to calculate the new expiration date and time. RETPD(0) causes the default retention period to be used for the dataset as defined in the PARMLIB. RETPD cannot exceed the maximum retention period MAXRETPD specified for your installation in the DFSMSrmm EDGRMM*xx* parmlib member.

When you change the retention period for a data set record representing one part of a multivolume data set on volumes managed by the EXPDT retention method and retained by SET or VOLUME, DFSMSrmm updates the expiration date and time for all the data set records for the data set.

Changing the retention period for a data set on a volume managed by the EXPDT retention method and retained by FIRSTFILE has no effect on the expiration of a volume or multivolume set unless it is the first file of the volume or volume set.

RETPD is mutually exclusive with EXPDT.

SECLEVEL(*security_class*)
Specifies the data set's security class. The value one to eight characters and 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 327 for more information.

To reset the security classification for a data set, use the NOSECLEVEL operand.

STEPNAME(*create_stepname*)
Specifies the job step name of the job that created the data set. A step name is one-to-eight characters other than blank, comma, and semicolon. 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 that created the data set. A storage class name is one-to-eight characters other than blank, comma, and semicolon. 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.

TOTALBLKCOUNT(*count*)
Use this operand to add a block count for a complete tape data set. The total block count is the sum of all block counts for all volumes on which the data set resides. The minimum allowable decimal value is 0. The maximum allowable decimal value is 18446744073709551615.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours

- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

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.

VRSELEXCLUDE(NO | YES)
Use this operand to override DFSMSrmm VRSEL processing. You can specify this for any data set on a volume managed by the VRSEL retention method. If VRSELEXCLUDE(YES) is specified for a data set already retained as a vital record, its vital record attribute is reset and the retention date set to the current date. The data set VRSELEXCLUDE attribute is set to YES for all data sets on volumes managed by the EXPDT retention method.

When a data set spans volumes, you should set the VRSELEXCLUDE attribute for each data set record – one data set record for each of the volumes on which the data set resides.

Specify NO to ensure a data set is included in VRSEL processing.

Specify YES to exclude a data set from VRSEL processing.

Authorization requires either CONTROL access to STGADMIN.EDG.MASTER, or UPDATE access to STGADMIN.EDG.CD.VX

VRSELEXCLUDE can be abbreviated as VX.

WHILECATALOG(OFF | ON | UNTILEXPIRED)
Use this operand to change the WHILECATALOG setting for a data set. WHILECATALOG applies only to data sets on volumes managed by the EXPDT retention method.

Specify OFF to indicate that retention of this data set should not depend on its catalog status and that only the expiration date is to be taken into account.

Specify ON to indicate that the data set should not be expired as long as it is cataloged, nor should it be expired prior to the expiration date. Changing the WHILECATALOG setting to ON does not affect the expiration date.

Specify UNTILEXPIRED to indicate that the data set should expire after its status changes to uncataloged or if the expiration date is reached.

When WHILECATALOG(ON) is specified, the data set expiration date is displayed with **(KeptByCatlg)** appended to the date, if the data set is cataloged. When WHILECATALOG(UNTILEXPIRED) is specified, the data set expiration date is displayed with **(OrUncatlg)** appended to the date, if the data set is cataloged.

Note: An uncatalog event for a data set that has WHILECATALOG(UNTILEXPIRED) specified could cause the expiration date of the data set to decrease. If the data set expiration date decreases and the current volume or volume set expiration date was copied from the expiration date of the data set being uncataloged, DFSMSrmm will assign a new expiration date for the volume or volume set. DFSMSrmm will scan through all of the data sets on the volume or volume set and find the maximal expiration date, which will then be assigned to the volume or volume set.

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 2012/001. 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 12001. 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 you want to set a date in the future, the FORCE operand is required.

  If a vital record specification or a data set LASTREF attribute indicates that DFSMSrmm retains 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. You can specify this operand if you are authorized to use the FORCE operand.

Task: Update the dataclass, managementclass, and storageclass information for a data set named DSBA063.DSN01.

Command:

```
RMM CHANGEDATASET 'DSBA063.DSN01' VOLUME(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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

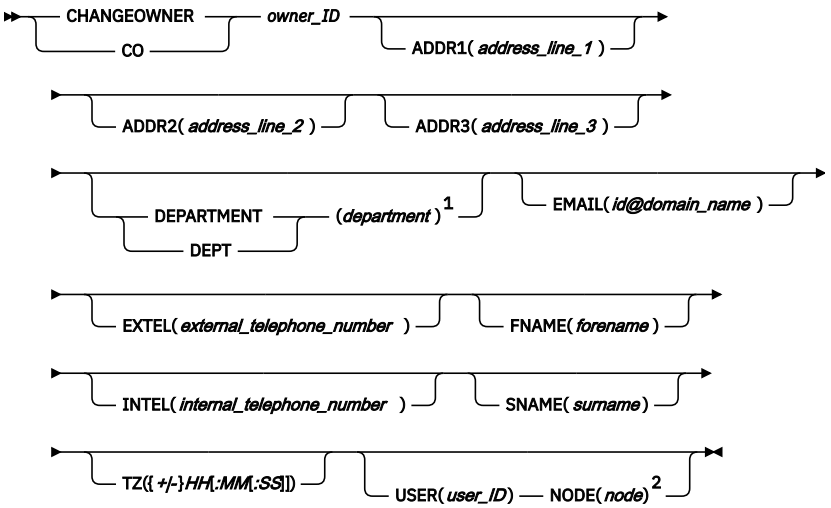
Purpose

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.

Format



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.

Parameters

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.

EMAIL(*id@domain_name*)
Specifies an Internet identifier for the user represented by this owner record. An Internet identifier specifies the path address of the mail recipient. The format of recipient is equivalent to the path syntax, as described in RFC 821, without the (<) and (>) delimiters. See *z/OS Communications Server: IP User's Guide and Commands* for details.

This EMAIL operand has one of these formats:

- user_id@host_name.domain - User on a host in a specified domain.
- user_id%nje_host_name@ gateway_name.domain - User on an NJE or RSCS node connected to a TCP network at gateway_name.

There is no default value. When you specify EMAIL, the USER and NODE values are ignored by NOTIFY processing, and any notify messages are sent using your system's SMTP server.

The maximum length of the value supported by DFSMSrmm is 63 characters.

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.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+/-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

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.

Task: Change these 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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

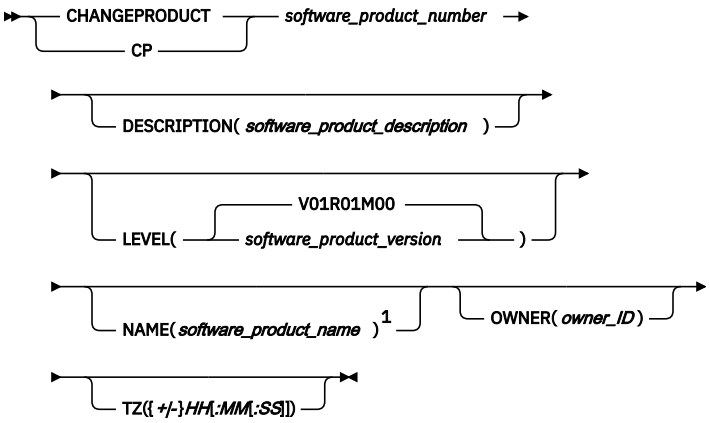
Purpose

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 269 and "DELETEVOLUME: Deleting volume information" on page 314 for more information.

Format



Notes:

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

Parameters

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 alphanumeric or national characters.

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 373 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.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

Examples

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose of CHANGEVOLUME subcommand

Before you begin:

1. 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 32:
     - 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 are described in "Changing volume information recorded by DFSMSrmm" on page 33 and"Reclaiming volumes from pending status or scratch status" on page 34:
     - You need CONTROL access to the STGADMIN.EDG.MASTER resource profile.
     - Depending on the security roles you have implemented to set and confirm moves and actions, if you do not have CONTROL access to STGADMIN.EDG.MASTER, you need UPDATE access to any or all of these profiles:
       - STGADMIN.EDG.MOVES.<location>.<destination>
       - STGADMIN.EDG.CMOVE.<location>.<destination>
       - STGADMIN.EDG.ACTIONS.<action>
       - STGADMIN.EDG.CRLSE.<action>
       - STGADMIN.EDG.CV.RM
   - 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 31 for information about using the DFSMSrmm ISPF dialog to change information for multiple volumes in a multivolume set.
- See "Rules for changing volume information" on page 32 for a summary of the rules for changing volume information with the RMM CHANGEVOLUME subcommand.
- See "Reclaiming volumes from pending status or scratch status" on page 34 for information on using the RMM CHANGEVOLUME subcommand to reclaim volumes from pending status or scratch status.
- See "Changing volume location" on page 35 for information on using the RMM CHANGEVOLUME subcommand change volume location.
- 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.

Use the MEDINF operand to assign installation-defined media information in DFSMSrmm parmlib member EDGRMMxx to the volume. To reset the volume to use the internal table for IBM media, use the following command: CV *volser* MEDINF(IBM).

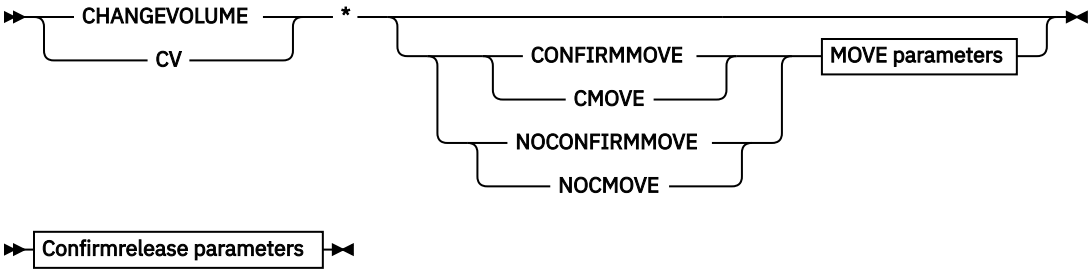
Provide a new rack number to change the volume's shelf location in the removable media library. The new rack number empty, or available for use. The old rack number is changed to EMPTY status. You also need to replace the external label on the volume to reflect the volume's new shelf location.

Supply a new owner ID to transfer ownership of the volume.

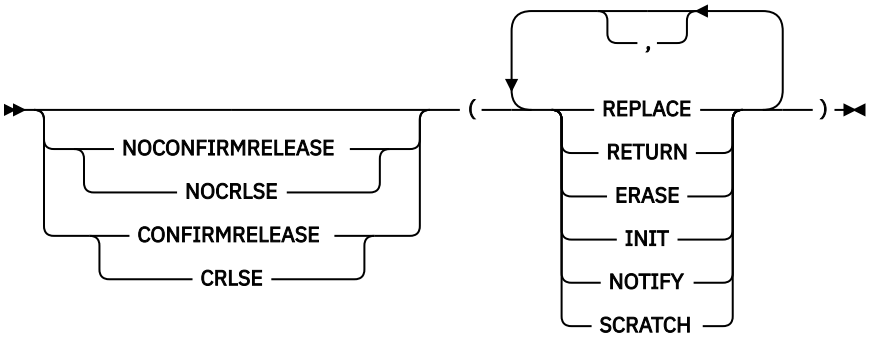
If you change the data set name, DFSMSrmm deletes all information about data sets on this volume and creates a new skeleton entry. Use the CHANGEDATASET subcommand to supply full data set details. See "CHANGEDATASET: Changing data set information" on page 251 for more information.

Format of CHANGEVOLUME global movement and confirm release operands

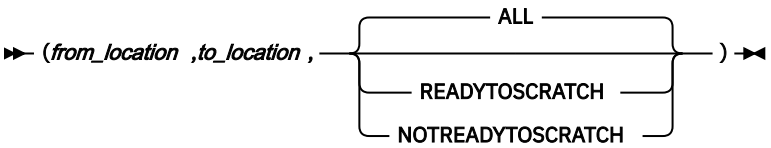
CHANGEVOLUME Global Movement and Confirm release Operands



Confirmrelease parameters

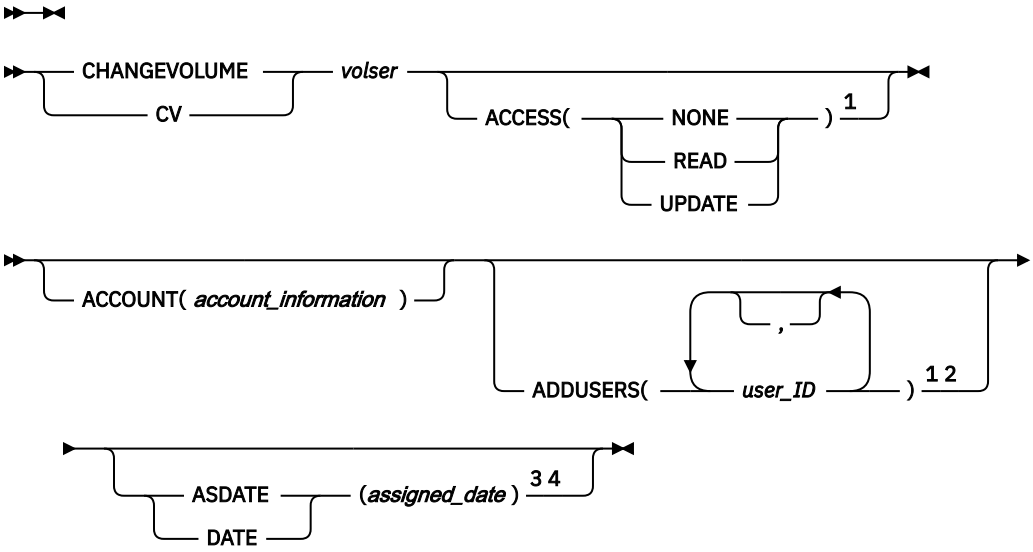


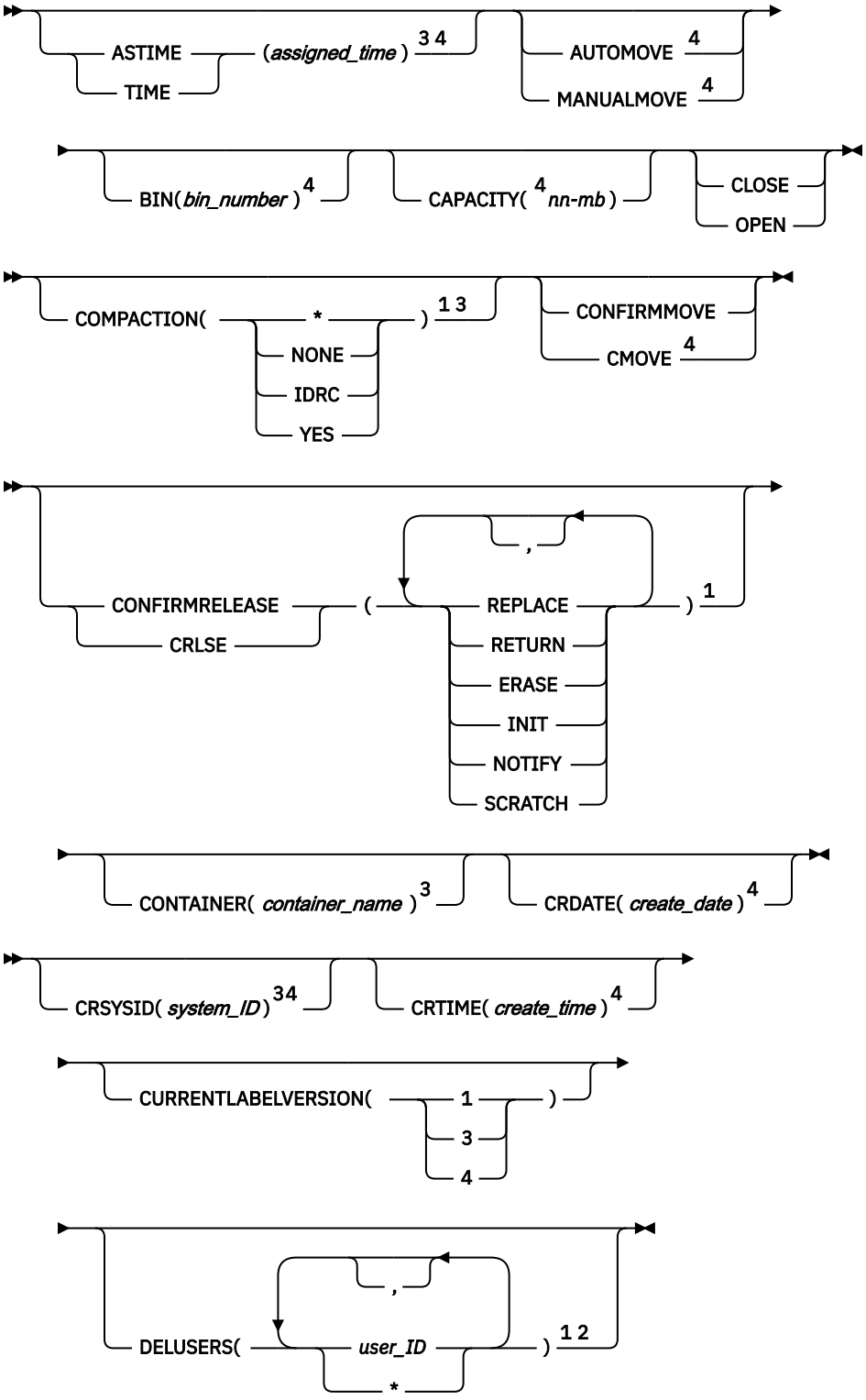
MOVE parameters

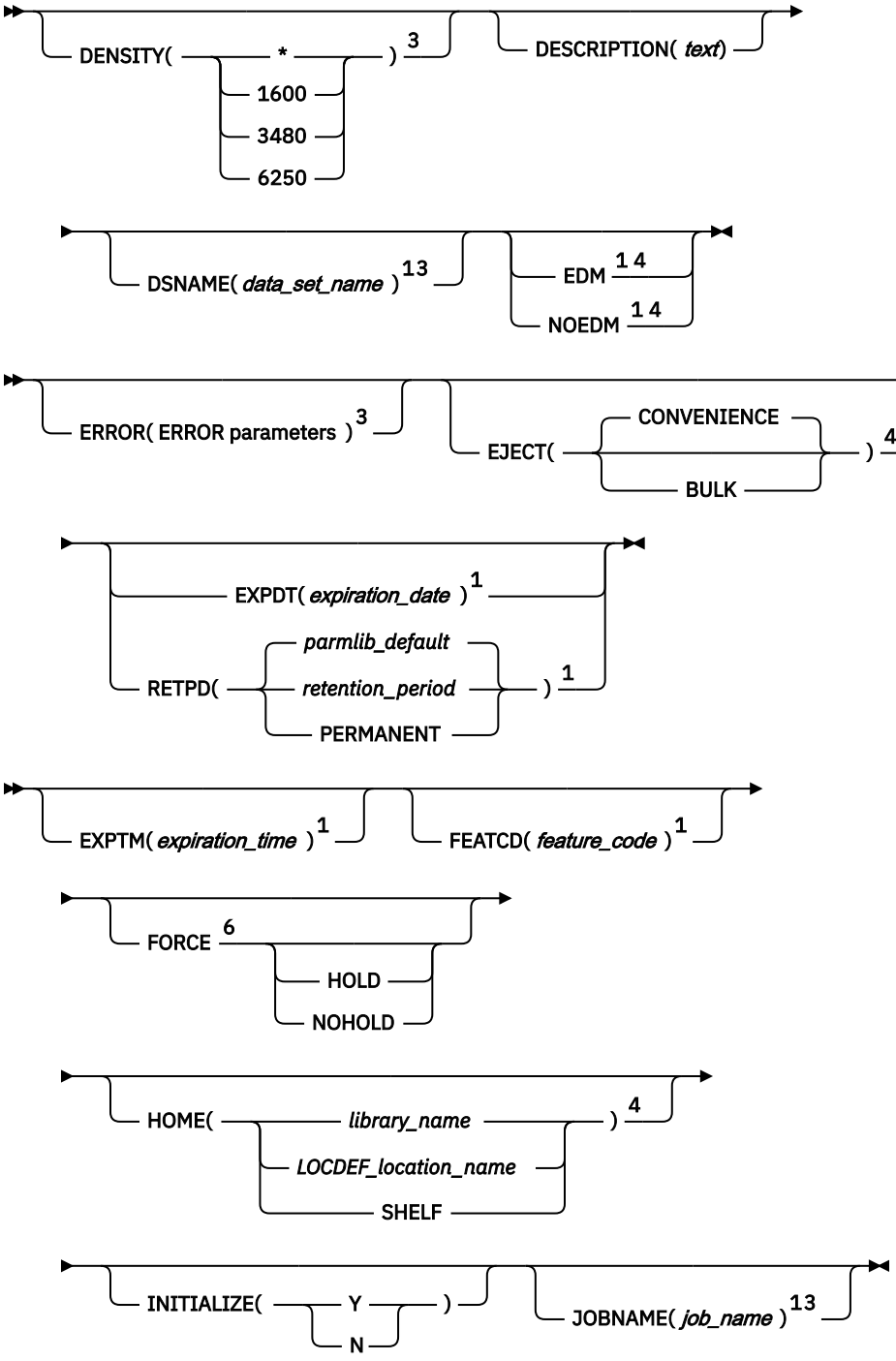


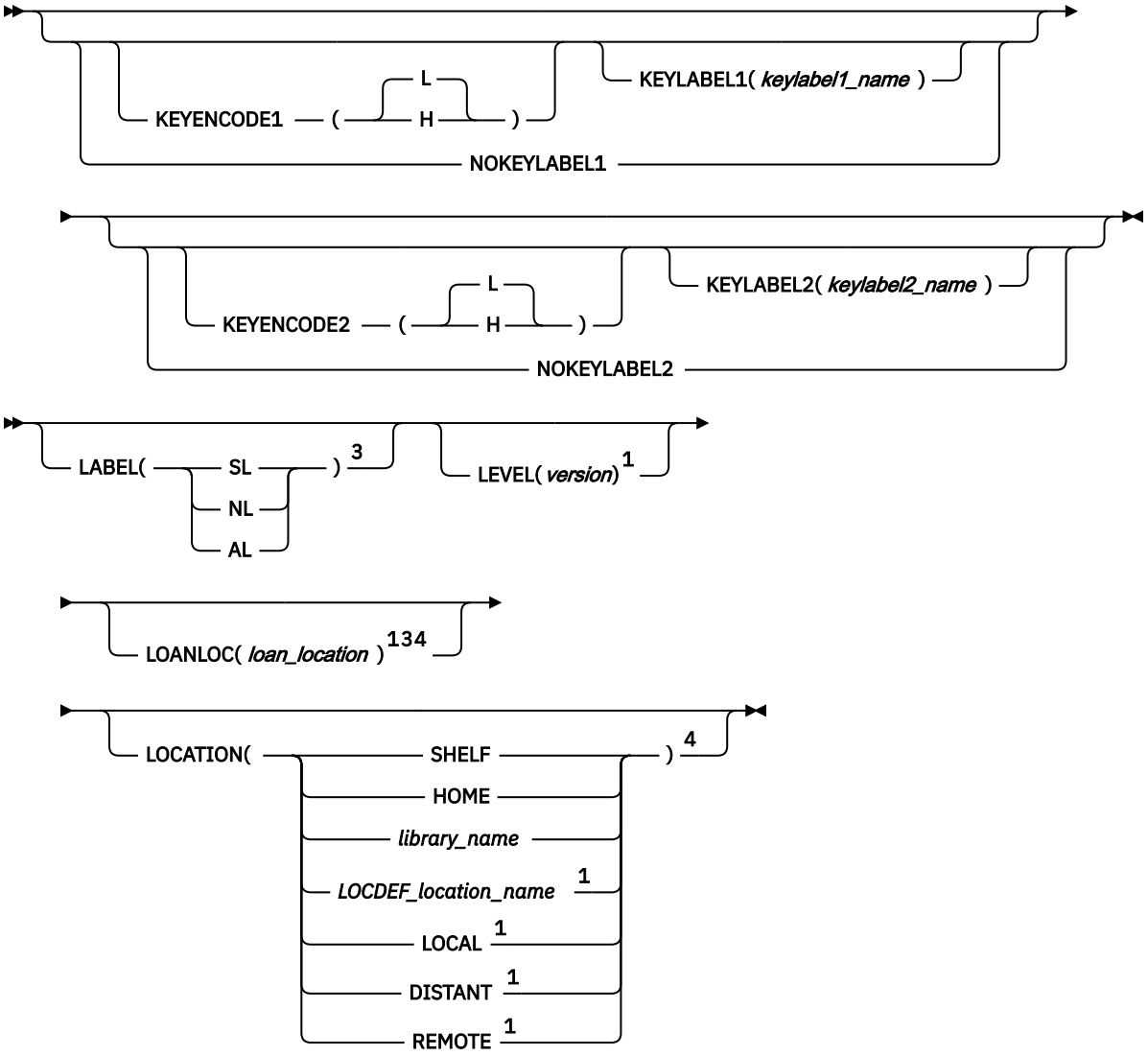
Format of CHANGEVOLUME specific volume operands

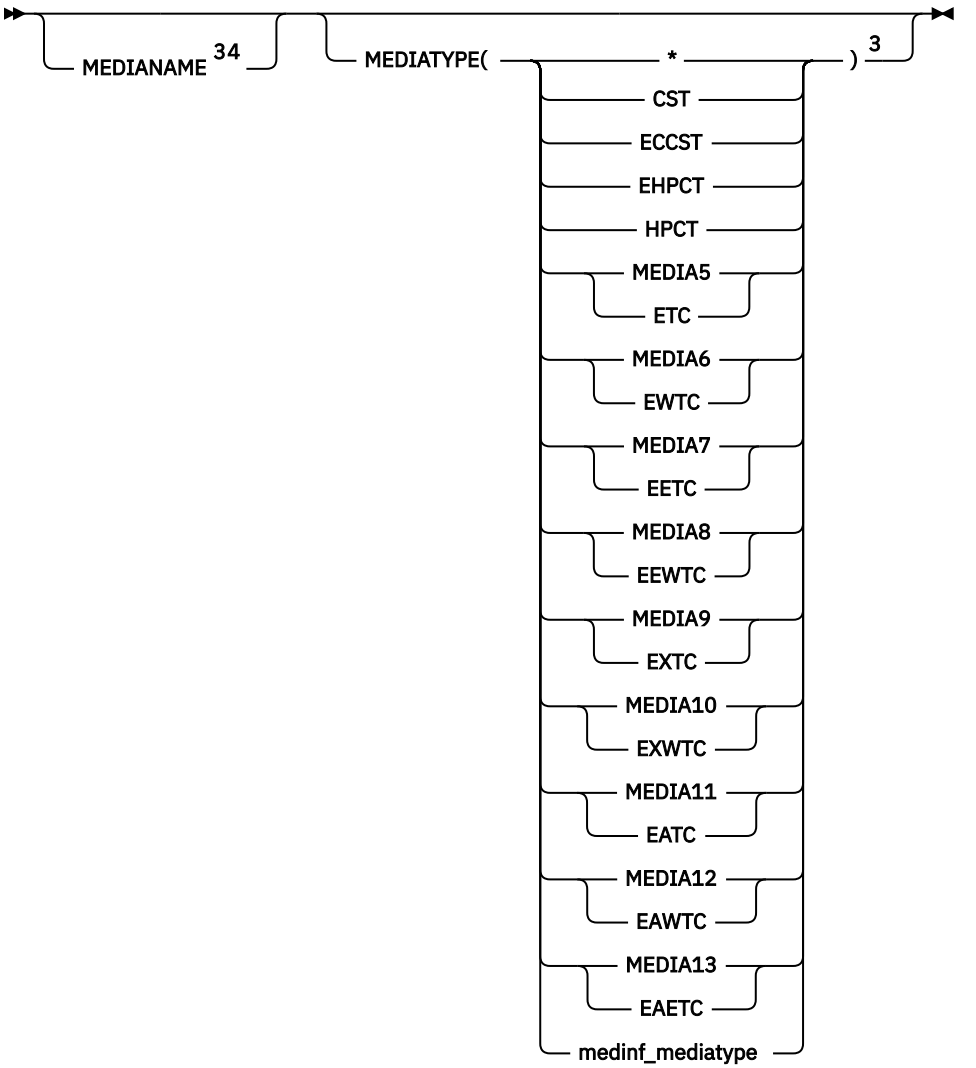
CHANGEVOLUME Specific Volume Operands

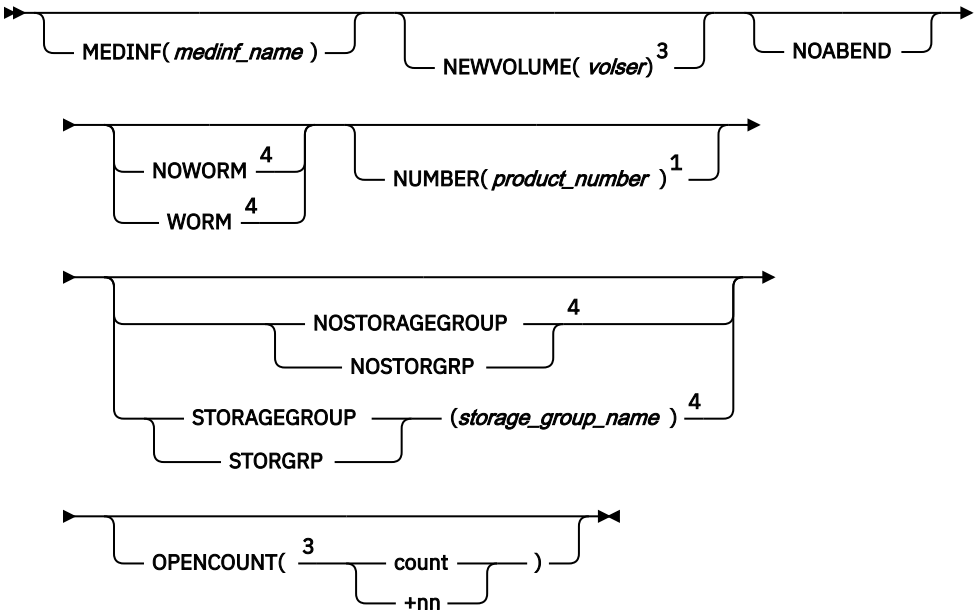


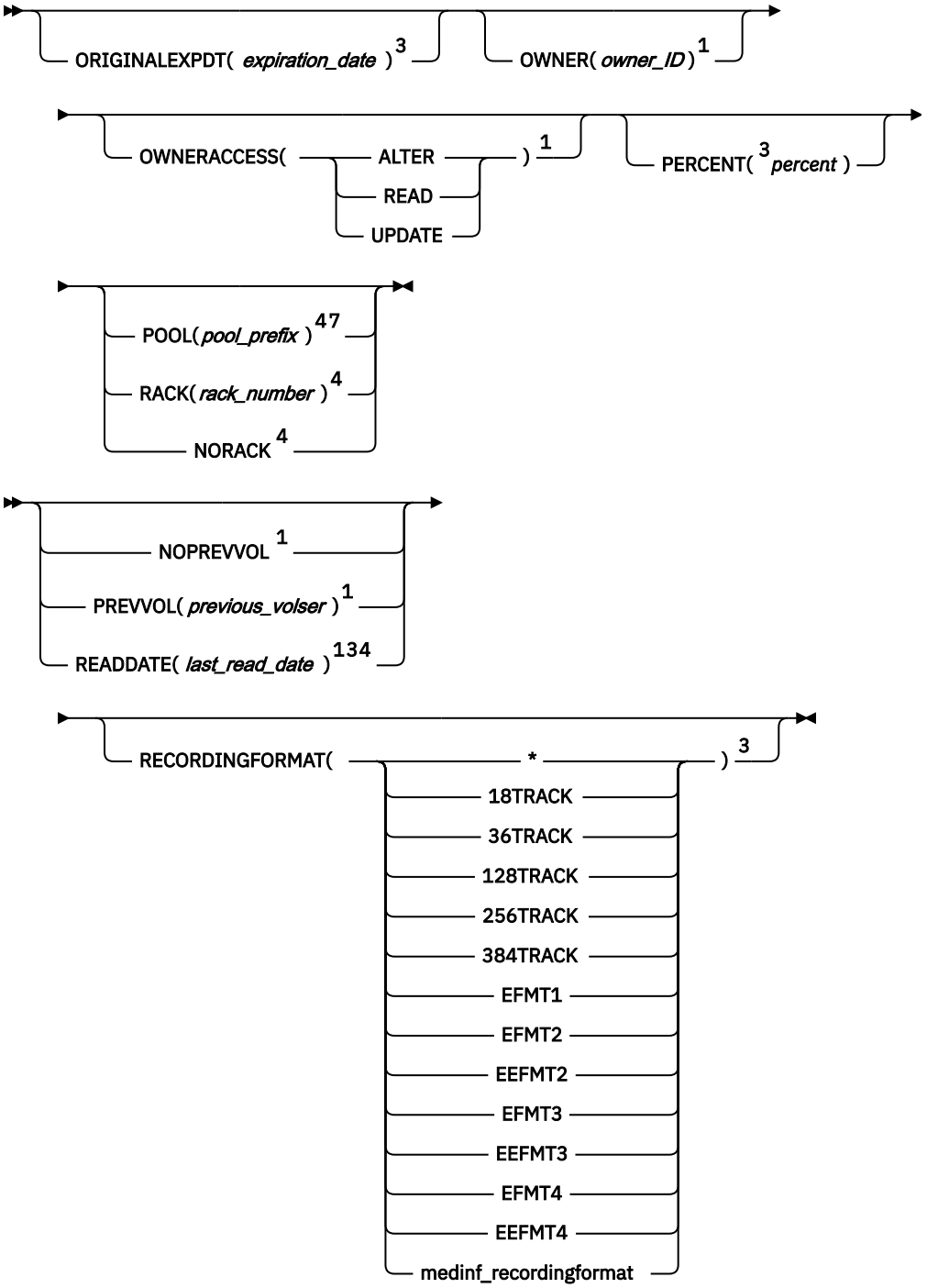

ASTIME
TIME
(assigned_time) 3 4
AUTOMOVE 4
MANUALMOVE 4
BIN(bin_number) 4
CAPACITY(4 nn-mb)
CLOSE
OPEN
COMPACTION(*) 1 3
NONE
IDRC
YES
CONFIRMMOVE
CMOVE 4
CONFIRMRELEASE
CRLSE
(REPLACE) 1
,
RETURN
ERASE
INIT
NOTIFY
SCRATCH
CONTAINER(container_name) 3
CRDATE(create_date) 4
CRSYSID(system_ID) 3 4
CRTIME(create_time) 4
CURRENTLABELVERSION(1)
3
4
DELUSERS(user_ID) 1 2
,
*

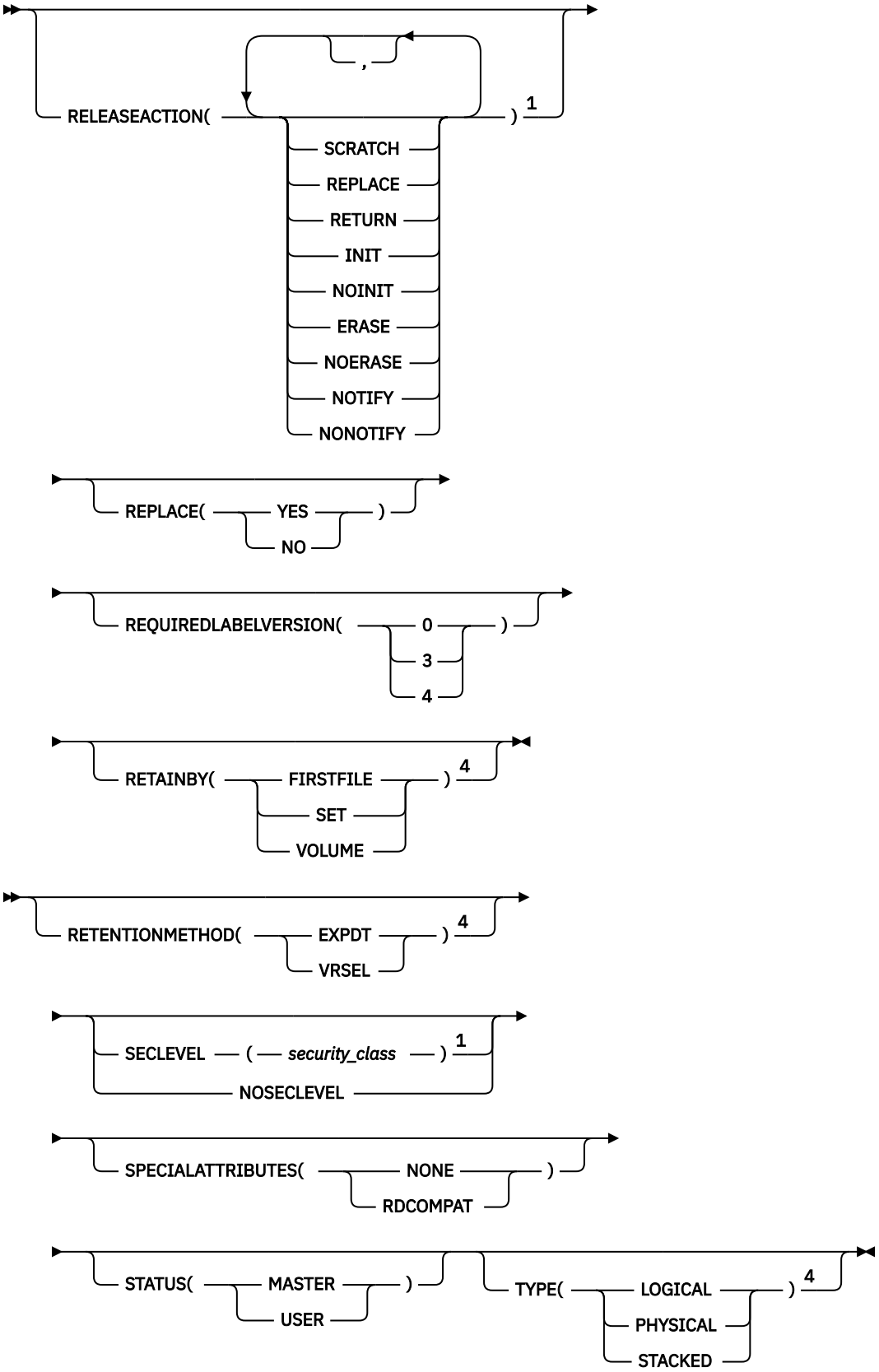

DENSITY(*) 3
1600
3480
6250
DESCRIPTION(text)
DSNAME(data_set_name) 13
EDM 14
NOEDM 14
ERROR(ERROR parameters) 3
EJECT(CONVENIENCE) 4 5
BULK
EXPDT(expiration_date) 1
RETPD(parmlib_default) 1
retention_period
PERMANENT
EXPTM(expiration_time) 1
FEATCD(feature_code) 1
FORCE 6
HOLD
NOHOLD
HOME(library_name) 4
LOCDEF_location_name
SHELF
INITIALIZE(Y)
N
JOBNAME(job_name) 13

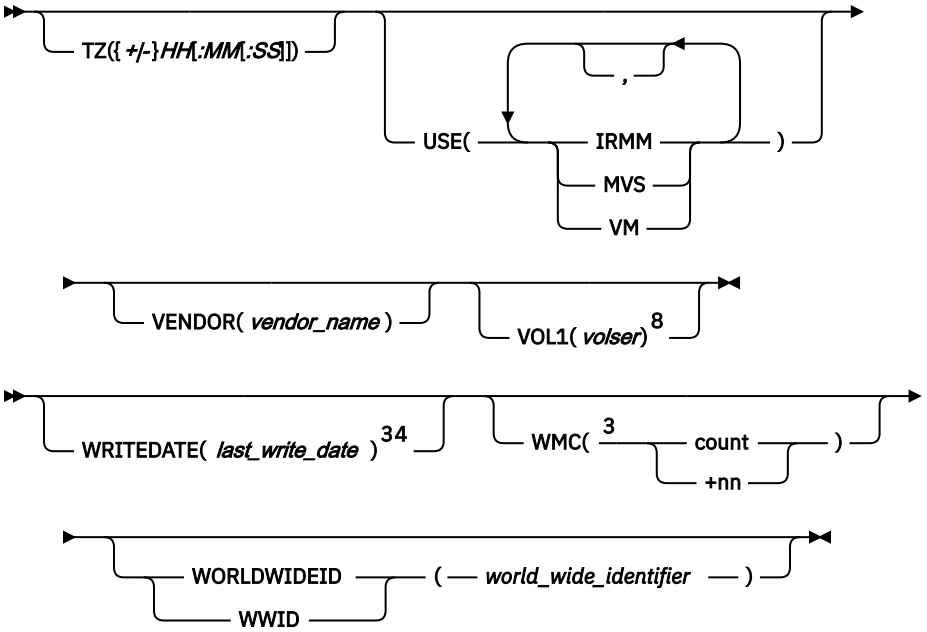

KEYENCODE1
(
L
H
)
KEYLABEL1(keylabel1_name)
NOKEYLABEL1
KEYENCODE2
(
L
H
)
KEYLABEL2(keylabel2_name)
NOKEYLABEL2
LABEL(
SL
NL
AL
) 3
LEVEL(version) 1
LOANLOC(loan_location) 134
LOCATION(
SHELF
HOME
library_name
LOCDEF_location_name 1
LOCAL 1
DISTANT 1
REMOTE 1
) 4


MEDIANAME 34
MEDIATYPE(*) 3
CST
ECCST
EHPCT
HPCT
MEDIA5
ETC
MEDIA6
EWTC
MEDIA7
EETC
MEDIA8
EEWTC
MEDIA9
EXTC
MEDIA10
EXWTC
MEDIA11
EATC
MEDIA12
EAWTC
MEDIA13
EAETC
medinf_mediatype

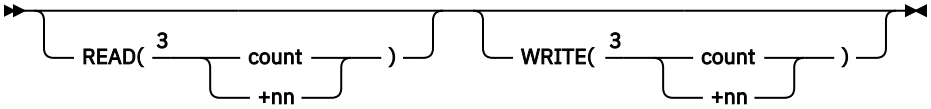

MEDINF(medinf_name)
NEWVOLUME(volser) 3
NOABEND
NOWORM 4
WORM 4
NUMBER(product_number) 1
NOSTORAGEGROUP 4
NOSTORGRP
STORAGEGROUP (storage_group_name) 4
STORGRP
OPENCOUNT(3 count)
+nn


ORIGINALEXPDT(expiration_date) 3
OWNER(owner_ID) 1
OWNERACCESS(ALTER READ UPDATE) 1
PERCENT(3 percent)
POOL(pool_prefix) 47
RACK(rack_number) 4
NORACK 4
NOPREVVOL 1
PREVVOL(previous_volser) 1
READDATE(last_read_date) 134
RECORDINGFORMAT(*) 3
18TRACK
36TRACK
128TRACK
256TRACK
384TRACK
EFMT1
EFMT2
EEFMT2
EFMT3
EEFMT3
EFMT4
EEFMT4
medinf_recordingformat


RELEASEACTION(
,
)
1
SCRATCH
REPLACE
RETURN
INIT
NOINIT
ERASE
NOERASE
NOTIFY
NONOTIFY
REPLACE(
YES
NO
)
REQUIREDLABELVERSION(
0
3
4
)
RETAINBY(
FIRSTFILE
SET
VOLUME
)
4
RETENTIONMETHOD(
EXPDT
VRSEL
)
4
SECLEVEL
(
security_class
)
1
NOSECLEVEL
SPECIALATTRIBUTES(
NONE
RDCOMPAT
)
STATUS(
MASTER
USER
)
TYPE(
LOGICAL
PHYSICAL
STACKED
)
4



ERROR parameters



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 value 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.

Parameters of CHANGEVOLUME subcommand

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 these:

NONE

Users do not have access to the volume

READ

Users only have read access

UPDATE
Users have write access to the volume

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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 cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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 2012/001. 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 12001. 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(*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.

ASTIME can be abbreviated as TIME.

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 is to be moved in a storage location. To assign a specific bin number, the volume's current location 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.

DFSMSrmm ignores this operand for scratch volumes if you do not specify either STATUS(MASTER) or STATUS(USER).

CAPACITY(*nn-mb*)
Use this operand to specify the volume capacity in megabytes (MB). DFSMSrmm normally sets the capacity of a volume based on the media type and the recording format, or you can manually set the capacity if the capacity cannot be determined from the media type and recording format. The MEDINF parmlib commands define the capacity of different combinations of media type and recording formats. There are built-in capacity values for IBM media types.

Specify a value between 0 and 4294967295.

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 these values:

*****
The compaction is not known; or the volume is not a tape volume, and compaction does not apply. This value 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 that are 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.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS 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 that are 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 these values for *from_location* and *to_location*:

ALL
All locations.

library_name
A shelf location in a system-managed library. Library names 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 327 for information about 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 performed an action for all volumes awaiting the indicated action.

Use CONFIRMRELEASE with one of these 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 cleanup activities necessary before returning the volume to scratch status. The necessary activities depend on your environment and can include updating the TCDB, uncataloging data sets, and deleting RACF profiles. Failing to perform a needed action can lead to inconsistencies.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

CONTAINER(*container_name*)
Specifies the name of the stacked volume that containins 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. CRDATE can be abbreviated as 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 2012/001. 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 12001. 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(*systemID*)
Specifies the ID of the system on which the volume was created. Specify a unique system name one-to-eight characters long. CRSYSID can be abbreviated as SYSID.

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.

CRTIME can be abbreviated as TIME.

CURRENTLABELVERSION(1|3|4)
Specifies that the ISO/ANSI label version set for the volume is to 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 cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

DENSITY(*|1600|3480|6250)
Specifies the recording density of the volume. For a 3420 tape reel, this value 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 uncatalogs only the subsequent data sets on the volume.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these masks will not match to data sets with all uppercase characters.

DFSMSrmm does not check quoted data set names for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. Data set names without quotation marks must pass these data set naming rules:

- A data set name can have one or more qualifiers.
- Each qualifier is one to eight characters. The first character alphabetic (A to Z) or national (# @ $). The remaining seven characters can be either alphabetic, numeric (0 - 9), national, or a hyphen (-).
- Qualifiers are separated by a period.
- DFSMSrmm adds your TSO PROFILE PREFIX value as the high-level qualifier.
- The data set name must not include a member name.

This operand can be changed if you have authorization to use the FORCE operand.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

EDM
Use the EDM (External Data Manager) operand to mark the volume as EDM-managed. Setting the EDM attribute prevents the volume from being expired during DFSMSrmm expiration processing. Authorization requires CONTROL access to STGADMIN.EDG.MASTER. The EDM operand is only valid for non-scratch, non-pending release volumes.

The RETAINBY(SET) option does not consider the EDM attribute; each volume needs to be set to EDM-managed separately.

EJECT
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.

ERROR(READ(*count*),WRITE(*count*))
Use this operand to specify new error count values for read and write errors. These are assumed to be the permanent errors for the volume.

You can specify an absolute value or an incremental value. Incremental values are specified as follows: WRITE(+23). If the value specified causes the count to exceed 65 535, the maximum value of 65 535 is set.

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 2012/001. 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 12001. 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.

When you specify the expiration date for a volume that is a part of a multivolume set that is managed by the EXPDT retention method and retained by SET, DFSMSrmm updates the expiration date and time for all the volumes of the multivolume set. When you specify the expiration date for a volume that is managed by the EXPDT retention method and retained by FIRSTFILE, the expiration date will be ignored.

EXPDT is mutually exclusive with RETPD. Changing the expiration date does not change the expiration time of the volume.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

EXPTM(*expiration_time*)
Specifies the volume expiration time, which is the time when the volume should be considered for release. The format for *expiration_time* is *hhmmss*, where:

hh
is hours

mm
is minutes

ss
is seconds

FEATCD(*feature_code*)
Specifies the software product feature code. A feature code is one-to-four alphanumeric characters.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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.

HOLD | NOHOLD

HOLD
Use the HOLD operand to set the volume HOLD attribute. Setting the HOLD attribute prevents automatic expiration and also prevents use of the RMM DELETEVOLUME subcommand with the RELEASE operand. The HOLD operand is only valid for non-scratch, non-pending release volumes. The RETAINBY(SET) option does not consider the HOLD attribute; each volume is subject to hold independently. HOLD and NOHOLD are mutually exclusive. Authorization requires either CONTROL access to STGADMIN.EDG.MASTER, or UPDATE access to STGADMIN.EDG.CV.HOLD.*volser*

NOHOLD
Use the NOHOLD operand to reset the volume HOLD attribute. While the HOLD attribute is set it prevents automatic expiration and also prevents use of the RMM DELETEVOLUME subcommand with the RELEASE operand. The NOHOLD operand is only valid for volumes which have the HOLD attribute set. HOLD and NOHOLD are mutually exclusive. Authorization requires either CONTROL access to STGADMIN.EDG.MASTER, or UPDATE access to STGADMIN.EDG.CV.NOHOLD.*volser*

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 or national characters.

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 characters other than blank, comma, and semicolon. You cannot use a generic job name. Any job name you use specific. This operand can be changed if you have authorization to use the FORCE operand.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

KEYENCODE1(L|H)
Specifies the encoding mechanism used for KEYLABEL1.

L
Label

H
Public key hash

If KEYLABEL1 is specified and no encoding method is yet defined to DFSMSrmm for this volume, the default value for KEYENCODE1 is L.

KEYENCODE2(L|H)
Specifies the encoding mechanism used for KEYLABEL2.

L
Label

H
Public key hash

If KEYLABEL2 is specified and no encoding method is yet defined to DFSMSrmm for this volume, the default value for KEYENCODE2 is L.

KEYLABEL1(*keylabel1_name*)
Specifies the key encryption key label number 1 for a non-scratch volume that is encrypted. A key label is 1-to-64 characters with blanks padding the field on the right. A key label contains alphanumeric, national, or special characters with some additional characters also allowed. Enclose it in single quotation marks if it contains any blanks or special characters.

When a volume is rewritten in a non-encryption format, DFSMSrmm does not clear the encryption key label related fields. Instead, the fields continue to be displayed until the volume is reused from scratch, or a release action causes them to be cleared.

When you specify this optional operand, you can also specify a value for the key encoding mechanism using the KEYENCODE1 operand.

KEYLABEL1 is mutually exclusive with NOKEYLABEL1.

KEYLABEL2(*keylabel2_name*)
Specifies the key encryption key label number 2 for a non-scratch volume that is encrypted. A key label is 1-to-64 characters with blanks padding the field on the right. A key label contains

alphanumeric, national, or special characters with some additional characters also allowed. Enclose it in single quotation marks if it contains any blanks or special characters.

When a volume is rewritten in a non-encryption format, DFSMSrmm does not clear the encryption key label related fields. Instead, the fields continue to be displayed until the volume is reused from scratch, or a release action causes them to be cleared.

When you specify this optional operand, you can also specify a value for the key encoding mechanism using the KEYENCODE2 operand.

KEYLABEL2 is mutually exclusive with NOKEYLABEL2.

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 alphanumeric or national characters. Use the ADDPRODUCT subcommand to add a new version of a software product number already defined to DFSMSrmm.

The default value is V01R01M00, Version 1, Release 1, Modification 0, when NUMBER is specified and the LEVEL has not yet been set.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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 cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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 these 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 can specify a distributed library name only if the library is an IBM Virtualization Engine.

LOCDEF_location_name
To indicate that the volume should be stored in an installation defined storage location.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

Use LOCATION with CONFIRMMOVE to mark the volume move as complete. You can use the LOCATION operand to cancel a pending move by specifying the volume's current location.

MANUALMOVE
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 | MEDIA9 | MEDIA10 | MEDIA11 | MEDIA12 | MEDIA13 | *medinf_mediatype*)
Specifies the volume's physical media type. Use one of these:

*****
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 Enterprise Tape Cartridge

MEDIA6/EWTC
IBM Enterprise WORM Tape Cartridge 3592

MEDIA7/EETC
IBM Enterprise Economy Tape Cartridge 3592

MEDIA8/EEWTC
IBM Enterprise Economy WORM Tape Cartridge 3592

MEDIA9/EXTC
IBM Enterprise Extended Tape Cartridge 3592

MEDIA10/EXWTC
IBM Enterprise Extended WORM Tape Cartridge 3592

MEDIA11/EATC
IBM Enterprise Advanced Tape Cartridge

MEDIA12/EAWTC
IBM Enterprise Advanced WORM Tape Cartridge

MEDIA13/EAETC
IBM Enterprise Advanced Economy Tape Cartridge

medinf_mediatype
Specify a non-IBM media type if your installation definition contains media information for *medinf_mediatype* that matches the media information assigned to the volume. When you change the media type, DFSMSrmm sets the volume capacity based on the matching media information.

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 28 for information about how DFSMSrmm assigns MEDIATYPE and RECORDINGFORMAT.

There is no default value.

MEDINF(*medinf_name*)
Specifies the assigned installation-defined media information to the volume. This value is one-to-eight alphanumeric characters and defined once in your installation. You can use the LISTCONTROL subcommand with the MEDINF operand to display the media information. For more information about MEDINF, see *z/OS DFSMSrmm Implementation and Customization Guide*. When you change the installation-defined media information, DFSMSrmm sets the volume capacity based on the matching media information.

Default: None.

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 these 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 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 these values to reverse confirmation of release actions performed manually for the volume being released. You can use one or more of these 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.

NOEDM
Use the NOEDM operand to reset the EDM attribute. EDM and NOEDM are mutually exclusive and can be applied to volumes with both EXPDT and VRSEL retention methods.

When the External Data Manager calls the EDGTVEXT or EDGDFHSM exit to release the tape, the EDM attribute is reset.

Be aware that resetting the EDM attribute for a volume that is still controlled by the External Data Manager may eventually cause data loss because DFSMSrmm will be allowed to release the volume earlier than expected.

NOKEYLABEL1
Specify the NOKEYLABEL1 operand to clear an existing key encryption key label number 1 and encoding mechanism.

NOKEYLABEL1 is mutually exclusive with KEYLABEL1.

NOKEYLABEL2
Specify the NOKEYLABEL2 operand to clear an existing key encryption key label number 2 and encoding mechanism.

NOKEYLABEL2 is mutually exclusive with KEYLABEL2.

NOPREVVOL
Specify to remove a single volume from the end of a multivolume chain.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

NOPREVVOL is mutually exclusive with RETENTIONMETHOD. To remove a volume from a volume set and also change the retention method requires two subcommands.

NORACK
Specifies to remove an assigned rack number from any volume. This allows the volume to be entered as a logical volume.

NOSECLEVEL
Specify to remove the security classification from a 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.

NOWORM
Use this operand to identify that the volume is not a WORM volume. You cannot specify this operand for a volume recorded automatically by DFSMSrmm during open processing.

Use the WORM operand to set the WORM attribute.

NOWORM is the default value.

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 cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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.

OPENCOUNT(*count* | *+nn*)
Use this operand to specify how many times any data set on the volume has been opened.

You can specify an absolute value or an incremental value. Incremental values are specified as follows: OPENCOUNT(+1). If the value specified causes the count to exceed 65535, the maximum value of 65535 is set.

ORIGINALEXPDT(*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 2007/001. 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 12001. 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.

`ORIGINALEXPDT` can be abbreviated as `OEXPDT`.

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 cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

OWNERACCESS(ALTER|READ|UPDATE)
Specifies the type of access the owner has to the volume.

When the RACF TAPEVOL class is active, and TPRACF(P) or TPRACF(A) is in effect, DFSMSrmm uses the OWNERACCESS information to build the RACF TAPEVOL access list. OWNERACCESS can be used together with OWNER to define the initial RACF TAPEVOL volume profile access, specifying the type of access the volume owner has to a volume.

The OWNERACCESS value can be one of these:

ALTER
The volume owner is allowed to read from the tape volume, to write add and delete data sets to the volume, and to create or destroy tape volume labels through OPEN or end-of-volume operations. For discrete tape volume profiles, the volume owner is allowed to change the profile, including the access list.

ALTER is the default value.

READ
The volume owner has only read access.

UPDATE
The volume owner is allowed to read from the tape volume, and to write additional data sets to the volume.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

For more information, refer to the topic Maintaining the user access list in *z/OS DFSMSrmm Implementation and Customization Guide*.

PERCENT(*percent*)
Use this operand to specify how full the volume is. You can specify a value between 0 and 100.

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 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.

The RETAINBY value for the added volume is taken from the previous volume and the expiration date will be updated, as required.

To remove a single volume from the end of a multivolume chain, specify PREVVOL('') or use the NOPREVVOL operand as shown in the example in "Examples of using the CHANGEVOLUME Command" on page 301.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

PREVVOL is mutually exclusive with RETENTIONMETHOD. To remove a volume from a volume set and also change the retention method requires two subcommands.

PREVVOL is mutually exclusive with RETAINBY.

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 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 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 2012/001. 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 12001. 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 want to set a date in the future, the FORCE operand is required.

This operand can be changed if you have authorization to use the FORCE operand.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

RECORDINGFORMAT(*| 18TRACK | 36TRACK | 128TRACK | 256TRACK | 384TRACK | EFMT1 | EFMT2 | EEFMT2 |EFMT3 | EEFMT3 | EFMT4 | EEFMT4 |*medinf_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. 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).

EFMT2
Data has been written to the volume in Enterprise Format 2 recording technology. You can only specify EFMT2 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), MEDIATYPE(MEDIA8), MEDIATYPE(MEDIA9), and MEDIATYPE(MEDIA10).

EEFMT2
Data has been written to the volume in Enterprise Encrypted Format 2 recording technology. You can only specify EEFMT2 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), MEDIATYPE(MEDIA8), MEDIATYPE(MEDIA9) and MEDIATYPE(MEDIA10).

EFMT3
Data has been written to the volume in EFMT3 (enterprise format 3) recording format. A recording format of EFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EEFMT3
Data has been written to the volume in EEFMT3 (enterprise encrypted format 3) recording format. A recording format of EEFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EFMT4
Data has been written to the volume in EFMT4 (enterprise format 4) recording format. A recording format of EFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) only.

EEFMT4
Data has been written to the volume in EEFMT4 (enterprise encrypted format 4) recording format. A recording format of EEFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) only.

medinf_recordingformat
Specify a non-IBM media recording format if your installation definition contains media information for *medinf_recordingformat* that matches the media information assigned to the volume. When you change the recording format, DFSMSrmm sets the volume capacity based on the matching media information.

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 28 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 these 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 these 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 this: `RELEASEACTION(SCRATCH,INIT,NOTIFY)`

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

REPLACE(YES|NO)
Use the REPLACE operand to change the setting of the replace action.

Specify YES to set on the REPLACE action when a volume is pending release with the SCRATCH release action, or if the volume is scratch. If a private volume is not pending release, you cannot change the REPLACE action (see instead the RELEASEACTION operand). DFSMSrmm processing sets off the pending SCRATCH action and sets the REPLACE action. DFSMSrmm processing for a scratch volume sets the volume to a master volume pending release with the REPLACE action pending. Any change from scratch to master for a system-managed volume requires access to the TCDB to update the volume status. Otherwise, the CV subcommand fails.

Specify NO to reset the replace action if it is a pending action. For pending release volumes, the SCRATCH action is set on by DFSMSrmm processing. This is not a confirmation that the REPLACE action is completed. Instead, use the CV subcommand with CRLSE(REPLACE) to confirm the REPLACE action. When you specify REPLACE(NO), you are avoiding the REPLACE action.

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.

RETAINBY(FIRSTFILE|SET|VOLUME)

Specifies how DFSMSrmm is to retain a volume or multivolume set managed by the EXPDT retention method. Specify this operand for the first volume in a multivolume sequence. All other volumes added to the set assume the same RETAINBY value.

FIRSTFILE
The expiration date of the first file is used to set the expiration date of a single volume or a multivolume set. All volumes in a multivolume set will have exactly the same expiration date and will be released to scratch in the same run of DFSMSrmm inventory management.

Additional data sets added later to a volume or to a multivolume set can have different expiration dates that are independent of the volume expiration date.

A cataloged data set with WHILECATALOG=ON will prevent the volume set from expiring only if it is the first file of the volume set.

SET
DFSMSrmm uses the highest expiration date of all volumes in the set and each file on a volume set can increment it. All volumes in the set will have exactly the same expiration date and will be released to scratch on the same run of DFSMSrmm inventory management.

Any cataloged data set with WHILECATALOG=ON will prevent the whole volume set from expiring.

VOLUME
The expiration date of the volume is considered for each volume separately and each file on a volume can increment the volume expiration date.

A cataloged data set with WHILECATALOG=ON will prevent expiration only of the volume that the data set is on.

Specify this operand for the first volume in a multivolume sequence. All other volumes added to the set will assume the same RETAINBY value.

Note:

1. The RETAINBY operand cannot be specified:
   - For a volume managed by the VRSEL retention method. Use the RETAINBY operand only for volume sets that use the EXPDT retention method.
   - For a SCRATCH volume.
   - If the PREVVOL operand is specified the volume.
2. When a RETAINBY value is defined for a non-scratch volume, it is not overridden to the default during OPEN output processing, but can be changed using CHANGEVOLUME subcommand.

Authorization requires either CONTROL access to STGADMIN.EDG.MASTER, or UPDATE access to STGADMIN.EDG.CV.RM

Default: If RETAINBY is omitted, the RETENTIONMETHOD(EXPDT(RETAINBY(*value*)) value specified in parmlib is used.

RETENTIONMETHOD(EXPDT | VRSEL)
Use this operand for any volume to set the retention method for a tape volume set. Specify this operand only for the first volume in a volume set. All other volumes in the set assume the same retention method.

Specify EXPDT to set the retention method for a tape volume set to be based on EXPDT. Data sets and volumes managed by this retention method are never processed by VRSEL inventory management. If the retention method is changed from VRSEL to EXPDT, then the RETAINBY field is set from the CV command, if specified, or from the parmlib. The expiration date of the volume or multivolume set is updated according to the RETAINBY value.

Specify VRSEL to set the retention method for a tape volume set to be VRSEL. This option enables DFSMSrmm inventory management to attempt to match data sets and volumes to vital record specifications, and if a match is found, to determine if the data set or volumes are to retained by VRS. If RETENTIONMETHOD(EXPDT) is changed to RETENTIONMETHOD(VRSEL), then the RETAINBY field for the volume is cleared.

When you change the retention method from VRSEL to EXPDT and the volume was retained as a vital record, the vital record attribute is reset and the retention date set to the current date. If the new retention method is the EXPDT retention method, the data set records will be excluded from VRSEL and the vital record attribute and retention date similarly changed. In addition, DFSMSrmm performs additional processing for all multivolume data sets in the volume set: the maximum expiration date and time for each multivolume data set is determined from the data set records and used to equalize the expiration for the multivolume data set.

RETENTIONMETHOD is mutually exclusive with PREVVOL and NOPREVVOL operands. To remove a volume from a volume set and also change the retention method requires two subcommands.

Authorization requires either CONTROL access to STGADMIN.EDG.MASTER, or UPDATE access to STGADMIN.EDG.CV.RM

RETENTIONMETHOD cannot be specified for a SCRATCH volume, unless the STATUS operand is also specified.

RETENTIONMETHOD can be specified as RM.

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 a decimal number from 0 to 93000 and is added to the current date and time to calculate the new expiration date and time. RETPD(0) causes the default retention period to be used for the volume as defined in the PARMLIB. RETPD cannot exceed the maximum retention period MAXRETPD set by your installation in parmlib member EDGRMM*xx*.

When you specify the retention period for a volume that is a part of a multivolume set that is managed by the EXPDT retention method and retained by SET, DFSMSrmm updates the expiration date and time for all the volumes of the multivolume set. When you specify the expiration date for a volume that is managed by the EXPDT retention method and is retained by FIRSTFILE, RETPD will be ignored.

RETPD is mutually exclusive with EXPDT.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

SECLEVEL(*security_class*)
Specifies a volume's security class. The value one to eight characters and 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 327 for more information.

To reset the security classification for a volume, use the NOSECLEVEL operand.

This operand cannot be specified for a SCRATCH volume, unless you specify the STATUS operand.

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 these:

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|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.

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.

For system-managed manual tape library volumes, the storage group name is used for scratch pool validation only when you request that a specific storage group name is used for pooling. For all non-system managed scratch pooling validation, the storage group name is significant and is always used to ensure that a volume from the correct pool is mounted.

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, which includes a non-IBM VTS that is being managed as a manual tape library.

TYPE(LOGICAL) is not allowed if the volume has a rack number that 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.

TZ(*{+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

USE(IRMM,MVS,VM)
Specifies the operating systems where the volume can be used. You can select one or more of: IRMM, MVS, and VM. When you change the systems where a volume can be used, include all the systems you need. To indicate multiple operating systems are valid, enter the values 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. Vendor information enclosed in single quotation marks if it contains any special characters or blanks. 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.

WMC(*count | +nn*)
Use this operand to set the write mount count for any volume. The write mount count reflects how many times the volume has been mounted for output and actually written to while mounted.

You can specify an absolute value or an incremental value. Incremental values are specified as follows: WMC(+1). If the value specified causes the count to exceed 65 535, the maximum value of 65 535 is set.

WORLDWIDEID|WWID(*world_wide_identifier*)
Use this operand to specify the unique world wide identifier set on the volume by the manufacturer. The world wide ID is 12 characters in hexadecimal. You can specify the value as 12 characters, a hex string of 24 characters (x'...'), or as a binary string (b'....'). When entered in hex, you specify 24 characters 0-9, A-F. An example of a world wide identifier is WWID(x'12345678ABCDEF090000FFEE'). When you specify a character string, it can be any string of 12 alphanumeric, national, special, or EBCDIC text characters, enclosed in quotes when special or EBCDIC text characters are specified. DFSMSrmm converts your character string to hex.

The WWID is maintained and displayed as a hexadecimal value and is displayed using 24 characters.

In some publications, the world-wide unique cartridge identifier (WWCID) can also be referred to as the world-wide identifier (WORLDWIDEID or WWID).

There is no default.

Do not use the DFSMSrmm subcommands to set or change the WWID value. Instead, add volumes as you normally would do, and DFSMSrmm records the value when the volume is first used. Subsequent use of the volume causes DFSMSrmm to ensure that the recorded WWID and the WWID obtained from the mounted volume both match. If the values do not match, the volume is rejected. Once the WWID is set by command or recorded by DFSMSrmm when the volume was used while mounted, you cannot change the value in the DFSMSrmm control data set. If the WWID is incorrect in DFSMSrmm, your choice is to either delete and then re-add the volume, or to use RMM REPLACE processing. If DFSMSrmm already has data set details for the volume, keep a record of these so they can be added back by using the DFSMSrmm subcommands.

WORM
Use this operand to identify that the volume is a WORM volume. When you set the WORM attribute and the volume is OPENed for input or output on z/OS, DFSMSrmm volume validation checks that the write mount count and worldwide unique volume ID values recorded in DFSMSrmm match those obtained from the cartridge and maintained by the drive.

Use the NOWORM operand to reset the WORM attribute.

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 2012/001. 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 12001. 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.

If you want to set a date in the future, the FORCE operand is required.

Examples of using the CHANGEVOLUME Command

Task: Remove the last volume in a multivolume chain.

Use this command:

```
RMM CHANGEVOLUME ABC123 PREVVOL('')
```

Or this command:

```
RMM CHANGEVOLUME ABC123 NOPREVVOL
```

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.

Use this command:

```
RMM CHANGEVOLUME ABC001 VOL1(ABC001)
```

Task: Delete these user IDs from the access list of volume 8E1U02: OWNER12, OWNER25, OWNER44 and OWNER45. Keep the volume for another 365 days.

Use this 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.

Use this 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.

Use this 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.

Use this 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.

Use this command:

```
RMM CHANGEVOLUME ABC123 NUMBER(1234-456) -
     FEATCD(1234) LEVEL(V01R01M00)
```

Task: If you use ISMF or the z/OS 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.

Use this command:

```
RMM CHANGEVOLUME ABC123 LOCATION(new_destination) CONFIRMMOVE
```

Return codes of CHANGEVOLUME subcommand

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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.

CHANGEVRS: Changing information about a vital record specification

Purpose

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

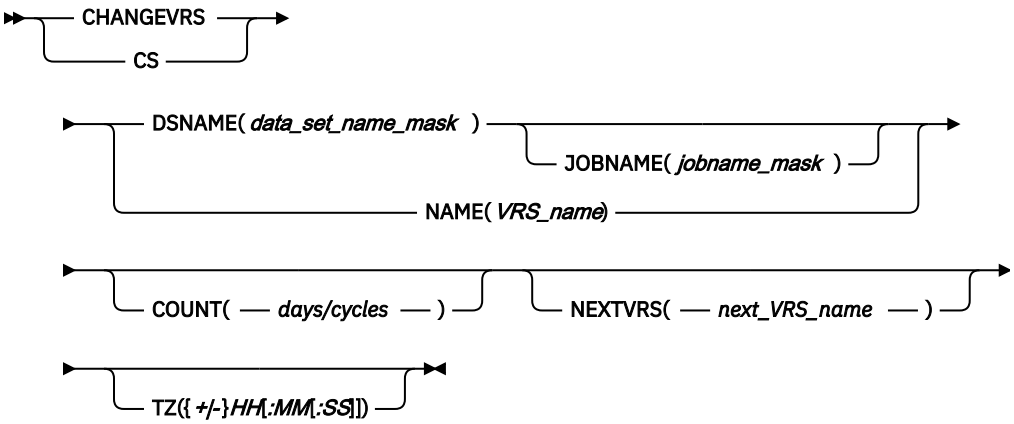
Use the CHANGEVRS subcommand to update details of a DSNAME or a NAME vital record specification that is already defined to DFSMSrmm.

When a vital record specification is changed, no data set or volume information is changed. During the next vital records processing run, DFSMSrmm uses the defined vital record specifications to apply policies. If the data set or volume matches to the changed vital record specification, DFSMSrmm applies the changed policies.

When you add, change, or delete a vital record specification, you should run VRSEL with the VERIFY option, as described in “Maintaining your vital record specifications” on page 109.

Use the SEARCHVRS subcommand to create lists of vital record specifications. See "SEARCHVRS: Creating a list of vital record specifications" on page 412 for more information.

Format



Parameters

DSNAME(*data_set_name_mask*)
Identifies the type of vital record specification and specifies the mask of a data set name, management class, or management value of an existing vital record specification. The mask can have a fully qualified or a generic name. It can also be one of the reserved words ABEND, DELETED, or OPEN.

The data set name mask is 1 to 44 characters, enclosed in quotes if any special characters are included. If the data set name mask is not enclosed in quotes, PROFILE PREFIX is applied. This operand is required and must immediately follow the CHANGEVRS subcommand.

DSNAME is mutually exclusive with the NAME and VOLUME operands.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

NAME(*VRS_name*)
Identifies the vital record specification type and specifies a name for the vital record specification. A vital record specification name is eight alphanumeric characters chosen by your installation.

JOBNAME(*jobname_mask*)
Identifies the job name for the vital record specification. A job name is one-to-eight alphanumeric characters, $, #, or @. You can specify a specific job name or a job name mask. It can also be one of the reserved words ABEND, DELETED, or OPEN.

This operand is optional. You must specify the operand, though, if the vital record specification you want to change has the JOBNAME operand specified.

If you have data sets with job names that include symbols other than alphanumeric characters, $, #, or @, use a job name mask to cover them.

COUNT(*days/cycles*)
Specifies a retention amount, based on the retention type of the existing vital record specification. Specify COUNT(number_of_days) to request that DFSMSrmm retains all cycles or copies of a data set. Specify COUNT(number_of_cycles) to request that DFSMSrmm retains the number of data set cycles you specify.

The value range for data set name vital record specification and retention name vital record specification is 0 to 99999. A value of 99999 indicates that DFSMSrmm retains all cycles of a data set.

If count() is not specified, the count value in the existing vital record specification record is not changed.

DFSMSrmm validates the COUNT value as follows:

- If EXTRADAYS is specified, COUNT must equal STORENUMBER: (STORENUMBER) = (COUNT).
- Regardless if NEXTVRS and ANDVRS are used, COUNT can be: (STORENUMBER) <= (COUNT).
- If DAYS or LASTREFERENCEDAYS retention is used: (STORENUMBER) + (DELAY) <= (COUNT).

NEXTVRS(*next_VRS_name*)
Specifies the name of the next vital record specification in a chain of vital record specifications. If you specify the name of a vital record specification that does not exist, DFSMSrmm uses a dummy vital record specification with the name '*broken*' instead. This keeps the volume or data set in its current location.

TZ({*+|-}HH[:MM[:SS]]*)
Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+|-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds

An optional colon (:) separates hours from optional minutes and optional seconds.

You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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 311 for the combined description of the DELETERACK and DELETEBIN operands.

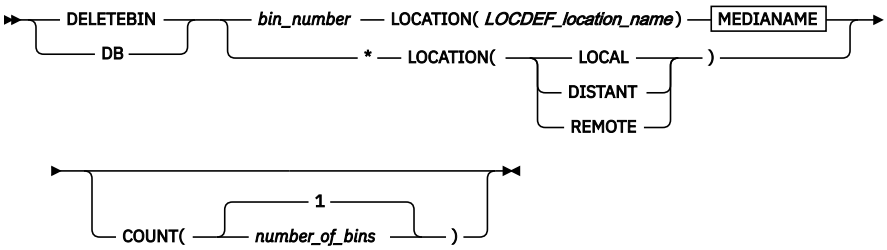
DFSMSrmm defines shelf space in storage locations as bin numbers. Use the DELETEBIN subcommand to delete information about shelf locations you are no longer using in built-in or installation defined storage locations.

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.

Format



MEDIANAME

Parameters

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 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.

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 10 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 11, “DFSMSrmm return codes and reason codes,” on page 423 for DFSMSrmm reason codes.

0
Subcommand completed normally.

4
Warning. Subcommand completed but some operands might 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

Purpose

Before you begin:

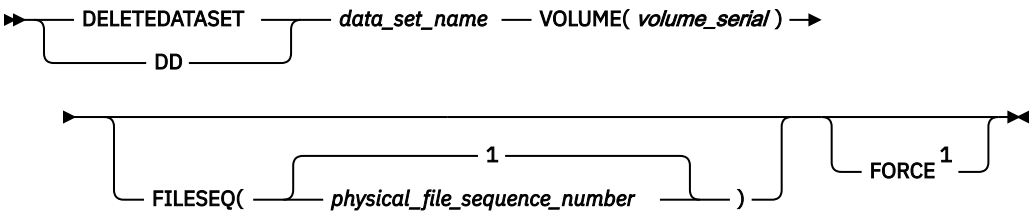
1. To use the RMM DELETEDATASET subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile.
2. To use the RMM DELETEDATASET FORCE subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile and UPDATE access to the STGADMIN.EDG.FORCE resource profile.

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.

Format



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.

Parameters

data_set_name
Specifies the name of the data set being deleted. The name is 1 to 44 characters in length and enclosed in quotes if any special characters are included. If the data set name is not enclosed in

quotes, your TSO PROFILE PREFIX value is applied. This operand is required and must immediately follow the DELETEDATASET subcommand.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

FILESEQ(*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.

FILESEQ can be abbreviated as SEQ.

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.

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

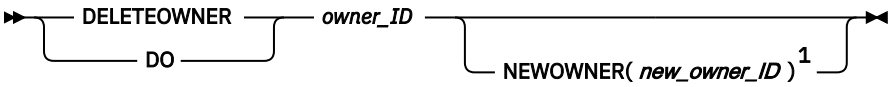
Purpose

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 314 for more information.

Format



Notes:

[1] The NEWOWNER operand specified to reassign volumes if the owner you are deleting owns one or more volumes.

Parameters

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.

Examples

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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.

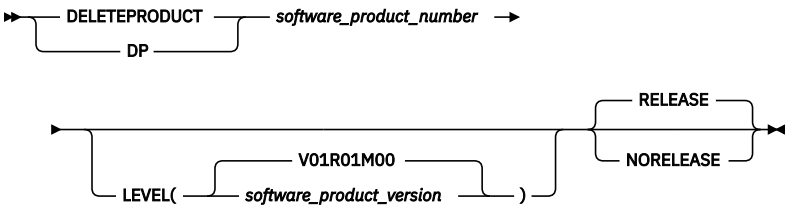
DELETEPRODUCT: Deleting software product information

Purpose

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.

Format



Parameters

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.

Examples

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 11, "DFSMSrmm return codes and reason codes," on page 423 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

Purpose

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

This topic describes the combined description for the DELETERACK subcommand and its alias DELETEBIN. See "DELETEBIN: Deleting bin number information" on page 305 for information about using the RMM DELETEBIN subcommand alias.

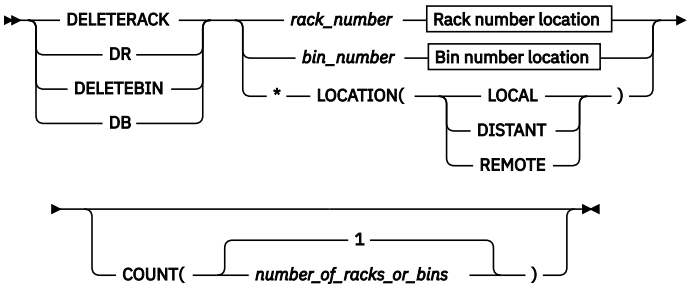
Use the DELETERACK subcommand to delete information about shelf locations you are no longer using in your removable media library or your storage locations. DFSMSrmm defines shelf space in the removable media library as rack numbers and bin numbers in storage locations.

See “DELETEBIN: Deleting bin number information” on page 305 for information about using the RMM DELETERACK subcommand alias.

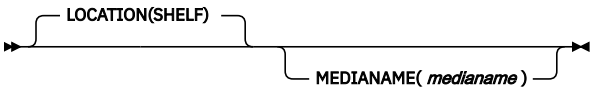
If you are deleting a specific rack number from the removable media library, you must specify a six-character rack number. If you are deleting more than one rack number, you must specify the initial rack number and a COUNT value. To delete rack numbers, DFSMSrmm uses the current VLPOOL definitions to 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.

Format



Rack number location



Bin number location

Parameters

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 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 empty.

A rack number, bin number, or an asterisk is required and must immediately follow the DELETERACK or DELETEBIN subcommand.

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

Before you begin:

- If you are a user, to use the RMM DELETEVOLUME RELEASE subcommand:
  - You need READ access to the STGADMIN.EDG.RELEASE resource profile to release your own volumes.
  - If 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.
- If you are a storage administrator, there are different authorization possibilities depending on the security roles you have implemented:
  - To use the RMM DELETEVOLUME subcommand, you need CONTROL access to the STGADMIN.EDG.MASTER resource profile. This allows all formats of the subcommand, except the use of the FORCE operand.
  - To use the RMM DELETEVOLUME subcommand with FORCE, you also need UPDATE access to the STGADMIN.EDG.FORCE resource profile.
  - To use the RMM DELETEVOLUME REMOVE subcommand, you need UPDATE access to the STGADMIN.EDG.DV.SCRATCH.volser profile.
  - To use the RMM DELETEVOLUME REPLACE subcommand, you need UPDATE access to the STGADMIN.EDG.CRLSE.REPLACE profile.

Related Reading:

- See “Deleting information for volumes in multivolume sets” on page 34 for information about using the DFSMSrmm ISPF dialog to delete volumes in a multivolume 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 four 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.

REPLACE
To delete information about a volume waiting to be replaced.

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.

Note: If the volume HOLD or EDM attribute has been set for the volume, DFSMSrmm will not release the volume.

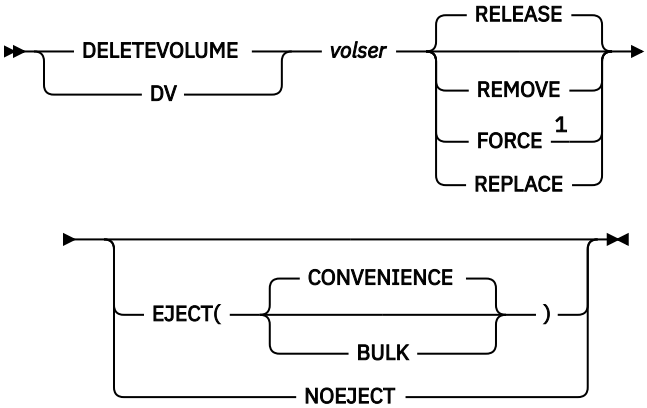
If you use the FORCE, REPLACE, or REMOVE operands to delete information about a volume, DFSMSrmm deletes information about any data sets on the volume. If the volume is in a system-managed tape library, you can also specify the EJECT operand to direct the volume to an exit station. For volumes that reside 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, REPLACE, or REMOVE operands. Specify NOEJECT with the FORCE, REPLACE, or REMOVE operands 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 the FORCE, REPLACE, or REMOVE operands for a volume in a system-managed tape library, DFSMSrmm purges information about the volume from the DFSMSrmm control data set based on the setting of SMSTAPE PURGE parmlib option. 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 113 for more information on releasing volumes.

Format



Notes:

[1] You must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource to use the FORCE operand.

Parameters

EJECT(CONVENIENCE|BULK)

Specifies directing a volume to an exit station, if you have also specified the FORCE, REPLACE, or 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 based on the setting of SMSTAPE PURGE parmlib option.

When specifying the FORCE, REPLACE, or REMOVE operands 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.

To use the FORCE operand, you must have CONTROL access to the STGADMIN.EDG.MASTER resource profile and UPDATE access to the STGADMIN.EDG.FORCE resource profile.

FORCE is mutually exclusive with RELEASE, REMOVE, and REPLACE. RELEASE is the default.

NOEJECT
Specifies preventing the volume from being ejected, if you also specified the FORCE, REPLACE, 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 only for logical and physical volumes for which the volume HOLD or EDM attribute has not been set.

RELEASE is mutually exclusive with REMOVE, FORCE, and REPLACE. 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 RELEASE, FORCE, and REPLACE. RELEASE is the default.

REPLACE
Specify to delete all information about a volume that is waiting to be replaced. REPLACE can only be specified when REPLACE is the only release action.

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 REPLACE to direct the volume to an exit station other than the convenience output station. During error recovery, you can use NOEJECT with REPLACE to prevent the volume from being ejected.

REPLACE is mutually exclusive with RELEASE, REMOVE and FORCE. 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.

Examples

Task: Delete information about the scratch volume with volume serial number 8E1U01.

Command:

```
RMM DELETEVOLUME 8E1U01 REMOVE
```

Task: Delete a volume if the SMS-managed library is no longer available.

Command:

```
RMM DELETEVOLUME volser NOEJECT FORCE
```

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

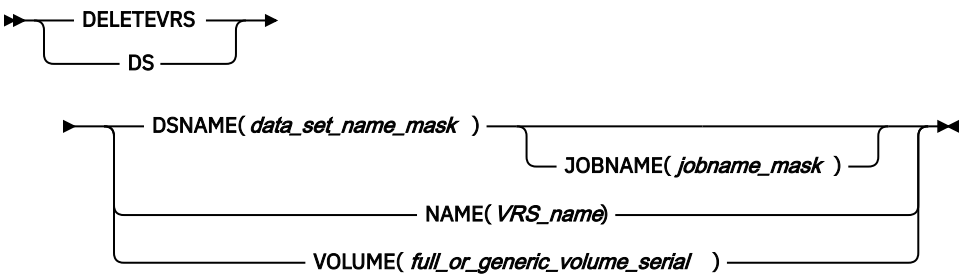
Before you begin: To use the RMM DELETEVRS subcommand, you need CONTROL access to the STGADMIN.EDG.VRS resource profile to delete vital record specifications.

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.

When you add, change, or delete a vital record specification, you should run VRSEL with the VERIFY option, as described in “Maintaining your vital record specifications” on page 109.

Format



Parameters

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. The name is 1 to 44 characters in length and enclosed in quotes if any special characters are included. If the data set name is not enclosed in quotes, your TSO PROFILE PREFIX value is applied. This operand is required and must immediately follow the DELETEVRS subcommand.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

You can also specify an SMS management class name, a vital record specification management value, or the reserved data set name masks,ABEND, DELETED, or OPEN. An SMS management class name or a vital record specification can be eight alphanumeric characters, beginning with an alphabetic character, and must follow standard z/OS 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.

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

Before you begin: To use the RMM GETVOLUME subcommand:

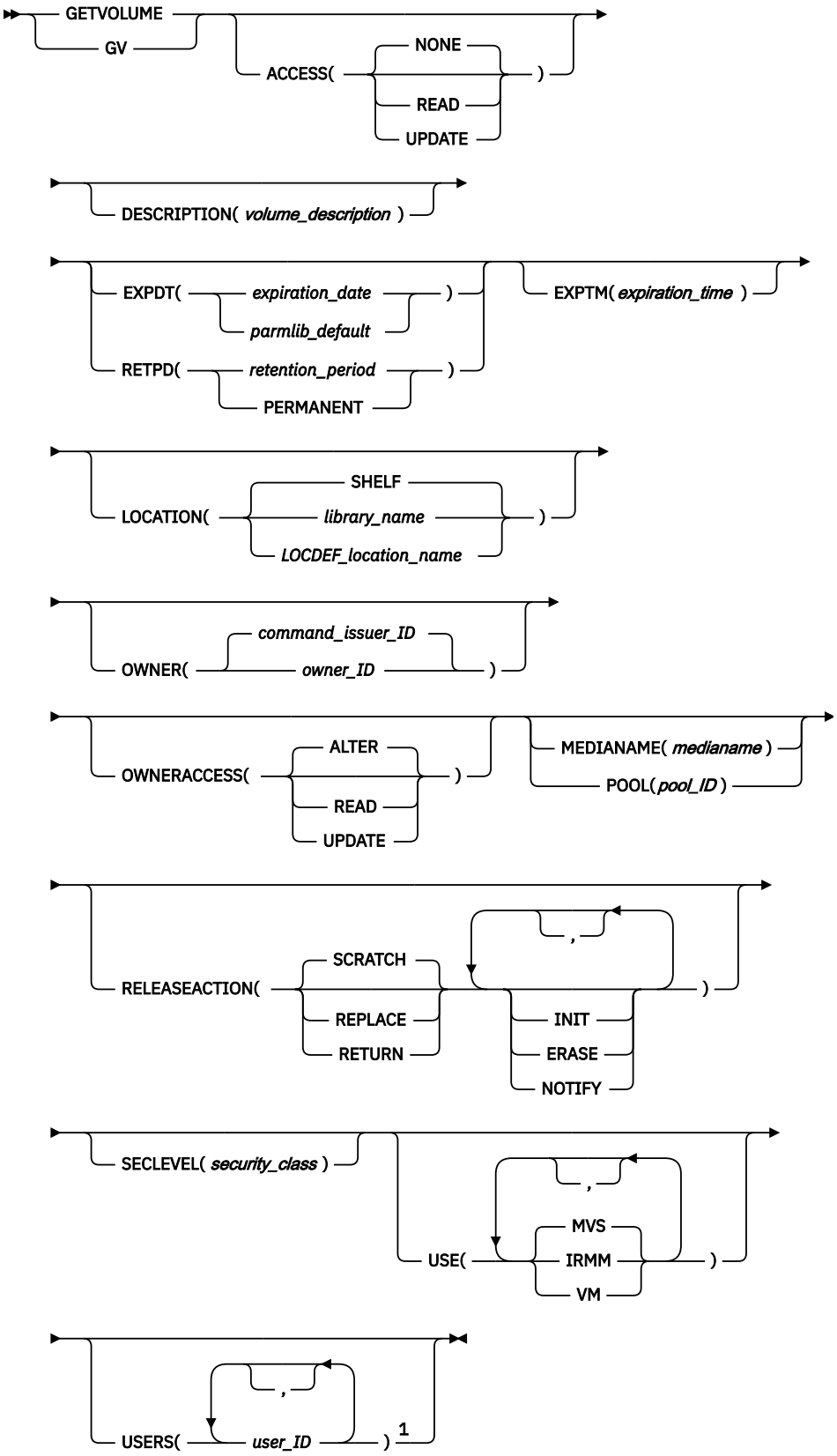
- You need READ access to the STGADMIN.EDG.MASTER resource profile to request a volume for yourself.
- You need CONTROL access to the STGADMIN.EDG.MASTER resource profile to request volumes for any user

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.

Format



Notes:

[1] You can specify a maximum of 12 user IDs.

Parameters

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 these:

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 2012/001. 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 12001. 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. Unless EXPTM is specified, the volume will be eligible for release on the midnight of the date that was provided in EXPDT.

EXPTM(*expiration_time*)
Specifies the time when the volume should be considered for release. The format for *expiration_time* is *hhmmss*, where:

hh
is hours

mm
is minutes

ss
is seconds

For example, nine o'clock in the morning is 090000. The expiration date is determined by the EXPDT or RETPD parameters. The time zone offset specified in the TZ options affects the resulting expiration time.

LOCATION(SHELF|*library_name*|*LOCDEF_location_name*)
Specifies a library from which the volume should be chosen. Specify one of these:

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

previously defined in the TCDB as a system-managed library. You can specify a distributed library name only if the library is an IBM Virtualization Engine.

LOCDEF_location_name
A *LOCDEF_location_name* is one-to-eight alphanumeric or national characters. It is the installation defined storage location name defined on LOCDEF in the current parmlib.

MEDIANAME(*medianame*)
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 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 327 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.

When the RACF TAPEVOL class is active, and TPRACF(P) or TPRACF(A) is in effect, DFSMSrmm uses the OWNERACCESS information to build the RACF TAPEVOL access list. OWNERACCESS can be used together with OWNER to define the initial RACF TAPEVOL volume profile access, specifying the type of access the volume owner has to a volume.

The OWNERACCESS value can be one of these:

ALTER
The volume owner is allowed to read from the tape volume, to write add and delete data sets to the volume, and to create or destroy tape volume labels through OPEN or end-of-volume operations. For discrete tape volume profiles, the volume owner is allowed to change the profile, including the access list.

ALTER is the default value.

READ
The volume owner has only read access.

UPDATE
The volume owner is allowed to read from the tape volume, and to write additional data sets to the volume.

For more information, refer to the topic Maintaining the user access list in *z/OS DFSMSrmm Implementation and Customization Guide*.

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 327 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 these:

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 these, 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 a decimal number from 0 to 93000. If PERMANENT is specified, the expiration date is set to 1999/365, and the volume never expires. The retention period is added to today's date and time to compute the EXPDT and the EXPTM. RETPD is mutually exclusive with EXPDT.

SECLEVEL(*security_class*)
Specifies the volume's security class. This value is one to eight characters and 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 327 for more information.

USE(IRMM,MVS,VM)
Specifies the operating systems where the volume can be used. You can select one or more of: IRMM, MVS, and VM. When you change the systems where a volume can be used, include all the systems you need and to indicate multiple operating systems are valid, enter the values 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.

Examples

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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

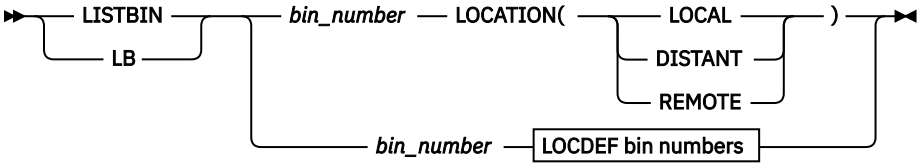
Before you begin: To use the RMM LISTBIN subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

The LISTBIN subcommand is an alias for the LISTRACK subcommand. See "LISTRACK: Displaying information about a shelf location" on page 342 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 351 for more information.

Format



LOCDEF bin numbers

Parameters

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 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.

Examples

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 75 on page 326:

```
Rack/bin number = M00001   Location = MAZBIN     Media name = 3480
Status          = IN USE                        Pool       =
Volume:
  Current       =
  Moving-in     = V10000
  Moving-out    =
  Old           = V10000

Last Change information:
Date             = 01/05/2011  Time = 02:29:53  System  = EZU0000
User change date = 01/05/2011  Time = 02:29:50  User ID = *HKP
```

Figure 75. Sample LISTBIN output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

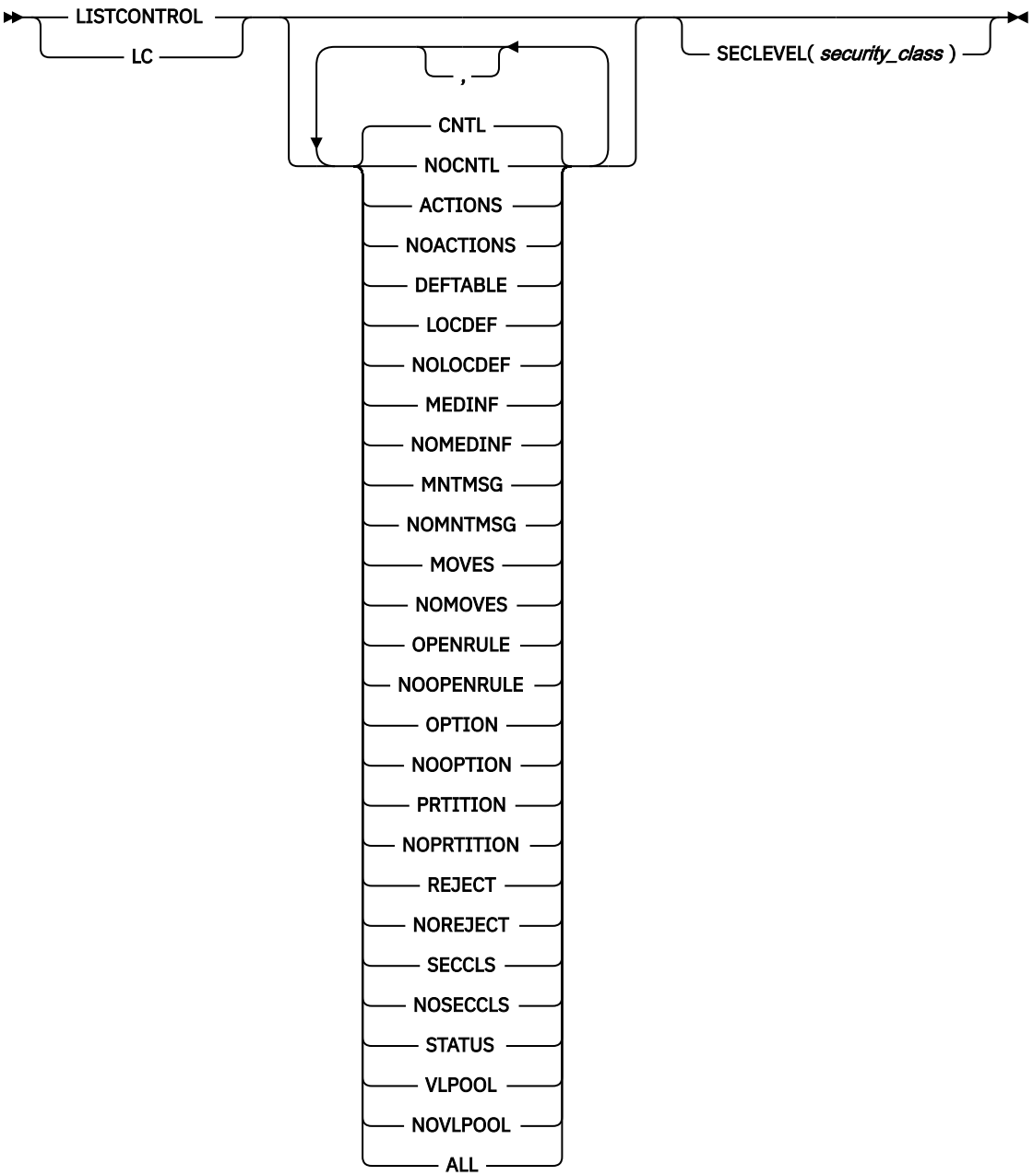
Before you begin: To use the RMM LISTCONTROL subcommand:

- 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 77 on page 331 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.

Format



Parameters

ACTIONS
Specifies to display information about outstanding volume actions. DFSMSrmm uses these 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, except the full defaults table. Specifying ALL is the same as specifying the operands ACTIONS, CNTL, LOCDEF, MEDINF, MNTMSG, MOVES, OPENRULE, OPTION, PRTITION, REJECT, SECCLS, and VLPOOL. You can use any of these operands with ALL to limit the amount of information displayed: NOCNTL, NOACTIONS, NOLOCDEF, NOMEDINF, NOMNTMSG, NOMOVES, NOOPENRULE, NOOPTION, NOPRTITION, 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.

DEFTABLE
Specifies to display content of table of defaults in use which is loaded from PARMLIB member of EDGDEFxx.

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.

MEDINF
Specifies to display the media information definitions that are defined to the installation in the EDGRMMxx parmlib member.

MNTMSG
Specifies to display the mount/fetch message definitions that are currently in use.

MOVES
Specifies to display information about outstanding volume moves. DFSMSrmm uses these 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.

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.

NOMEDINF
Specifies to prevent media information from being displayed when you specify LISTCONTROL ALL.

NOMNTMSG
Specifies to prevent the mount/fetch message definitions from being displayed when you specify LISTCONTROL ALL.

NOMOVES
Specifies to prevent outstanding volume moves from being listed when you specify LISTCONTROL ALL.

NOOPENRULE
Specifies to prevent the open rule information from being displayed when you specify LISTCONTROL ALL.

NOOPTION
Specifies to prevent system options from being displayed when you specify LISTCONTROL ALL.

NOPRTITION
Specifies to prevent the partitioning information 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.

OPENRULE
Specifies to display the open rule definitions that are currently in use.

OPTION
Specifies to display the system options that are currently in use.

PRTITION
Specifies to display the partitioning definitions 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 a security class. Specify a security class ID. The value one to eight characters and must correspond to one of the security classes defined for your installation.

STATUS
Specifies to display information about DFSMSrmm subsystem address space status, tasks, and queued requests. The information returned is very similar to the results of the operator 'F DFRMM,QUERY ACTIVE' command.

Note:

1. Although the outputs of LISTCONTROL STATUS and QUERY ACTIVE are very similar in terms of the kinds of information displayed, if the two commands are issued simultaneously, the output of the LISTCONTROL STATUS might reflect a slightly later state than the QUERY ACTIVE command, owing to different processing paths for the two commands.
2. The output of the LISTCONTROL STATUS command will always include at least one active task (LC) for itself.
3. The STATUS operand is not included with the ALL operand.

VLPOOL
Specifies to display information about pools defined for your installation.

Task: Display information about the DFSMSrmm subsystem, subsystem requests, and task status.

Command:

```
  RMM LISTCONTROL STATUS
```

Output: DFSMSrmm displays information such as that shown in Figure 76 on page 331:

```
RMM LC STATUS
 DFSMSrmm status = ACTIVE   Journal = ENABLED  Server listener = ACTIVE
 Local tasks     = 5        Server tasks    = 5
       Active    = 2               Active   = 2
       Held      = 1               Held     = 1
 Queued requests = 0        New requests    = NOTHELD
       Nowait    = 0
       Catalog   = 0
 Last RESERVE    = 06:17:09 + ENQ
 Debug Setting   = DISABLED
 Trace Levels    = 1,2

 Active requests:
 Function  System  Task   Name  Started   Token   S IP Status
 --------- -------- ------------ -------- -------- - ----------------
 LC                 JOB=RMMUSERS 06:17:09 00400009
 HSKP               JOB=INVMGMTS 05:29:27 00300002 H
 ADD       EZU34    JOB=RMMUSERS 06:15:49 0060000B + READ  < 06:17:09
 C/S       EZU34    STC=DFRMM    00:00:00 00700001   READ  > 06:16:52
```

Figure 76. Sample LISTCONTROL STATUS output

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 77 on page 331 and Figure 78 on page 332:

```
  Control record:
  Type = MASTER    Create date = 05/13/2015  Create time = 03:30:00
                   Update date = 05/13/2015  Update time = 03:30:07
  Journal: Utilization =   0% (50% threshold)     STATUS: = ENABLED
  CDS:     Utilization = 100%  (90% threshold)

  Exit status:                              Options:
    EDG_EXIT100 = NONE                        Stacked Volumes         = NONE
    EDG_EXIT200 = NONE                        Extended Bin            = DISABLED
    EDG_EXIT300 = NONE                        Common Time             = DISABLED
                                              CDSID ENQ name          = ENABLED
  Last backup:                              Last expiration processing:
    Date =             Time =                 Date =             Time =
  Last journal backup:                      Last store update:
    Date =             Time =                 Date =             Time =
  Last report extract:                      Last VRS processing:
    Date =             Time =                 Date =             Time =
  Last scratch procedure:                   Last Catalog synchronize:
    Date =             Time =                 Date =             Time =
  Rack numbers       = 0                    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
```

Figure 77. Sample LISTCONTROL CNTL OPTION output (Part 1 of 2)

```
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  =
  IP address =

System options:
PARMLIB Suffix  = $$           Operating mode  = P
Control data set name      = RMMUSER.APAR.MASTER
Journal file data set name = RMMUSER.APAR.JOURNAL
Journal threshold          = 50%              Journal transaction = NO
Catalog SYSID              = Notset
CATSYSID list              =



Scratch procedure name     = EDGXPROC
Backup procedure name      = SUB
IPL date check  = N        Date format    = A       RACF support   = N
SMF audit       = 0        SMF security   = 0       CDS id         = SJTEST
MAXHOLD value   = 250      Lines per page = 54      System ID      = W98MVS2
BLP             = RMM      TVEXT purge    = RELEASE Notify         = Y
                           days           = 0
Uncatalog       = S                                 Message case   = M
MASTER overwrite= MATCH    Accounting     = J
Disp DD name    =          Disp msg ID    = EDG4054I
PREACS          = NO       SMSACS         = NO      CMDAUTH OWNER  = YES
Reuse bin       = CONFIRMMOVE                       CMDAUTH DSN    = NO
Local tasks     = 10                                Media name     = 3480
EDM             = NO       DSNEXPIRE      = NONE
HouseKeeping    = LIMITED


Retention period: Default = 5     Maximum = NOLIMIT
                  Catalog = 1     hours
Use of Management Class Attributes = NONE
Retention method: Default = VRSEL



RM(VRSEL) defaults:              RM(EXPDT) defaults:
  Retain by         = VOLUME       Retain by             = VOLUME
  Move By           = VOLUME
  VRS selection     = NEW          LASTREF extra days    = 0
  VRS change        = INFO         Catalog Days          = 2
  VRSMIN action     = FAIL         GDG    : WHILECATALOG = OFF
  VRSMIN count      = 1                     RETPD        = 5
  VRS job name      = 2            Non-GDG: WHILECATALOG = OFF
  GDG duplicate     = BUMP                  RETPD        = 5
  GDG cycle by      = GENERATION

VRSDROP action  = INFO     VRSDROP count  = 0              percent = 10
VRSRETAIN action= INFO     VRSRETAIN count= 0              percent = 80
EXPDTDROP action= INFO     EXPDTDROP count= 0              percent = 10

PDA: ON
 Block count    = 255      Block size     = 27      Log            = ON
SMSTAPE:
 Update scratch = YES      Update command = YES     Update exits   = YES
 Purge          = ASIS
Client/Server:
 Subsystem type = STANDARD Port           = 0
 Server                    Server tasks   = 0
  host name  =
  IP address =

Defaults table definitions:
DEFTABLE is loaded from PARMLIB member EDGDEF16, number of entries: 3
To display the complete defaults table, issue the following command: RMM LISTCONTROL DEFTABLE
```

Figure 78. Sample LISTCONTROL CNTL OPTION output (Part 2 of 2)

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 79 on page 333.

```
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 79. Sample LISTCONTROL output - options and rules, excluding control record information and start up options.

Task: Display your installation's partitions.

Command:

```
RMM LISTCONTROL PRTITION
```

Output: DFSMSrmm displays information such as that shown in Figure 80 on page 333.

```
Partition Entries:
                        SMT       NOSMT
 Volume or Range Type   Action Action Location
 --------------- -----  ------ ------ --------
 AZ99B0          RMM    ACCEPT ACCEPT SHELF
 A00100:Z99AB0   RMM    IGNORE IGNORE
 *               RMM    IGNORE IGNORE
 AZ99B0          NORMM  ACCEPT ACCEPT SHELF
 B00001:B00001   NORMM  ACCEPT ACCEPT SHELF
 MW*             NORMM  ACCEPT ACCEPT LOC1
 *               NORMM  IGNORE IGNORE
```

Figure 80. Sample LISTCONTROL PRTITION output

Task: Display your installation's OPENRULE entries.

Command:

```
RMM LISTCONTROL OPENRULE
```

Output: DFSMSrmm displays information such as that shown in Figure 81 on page 334.

```
Openrule Entries:
                                Input                          Output
 Volume or Range Type  Action Condition   SAFRC8 Action  Condition   SAFRC8
 --------------- ----- ------ ----------- ------ ------  ----------- ------
 A00100:Z99AB0   RMM   REJECT                    REJECT
 *               RMM   REJECT                    ACCEPT
 B00001:B00001   NORMM REJECT                    REJECT SYSID,CATLG
 MW*             NORMM IGNORE SPECIFIC          REJECT
 *               NORMM ACCEPT                   ACCEPT
```

Figure 81. Sample LISTCONTROL OPENRULE 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 82 on page 334:

```
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 82. Sample LISTCONTROL SECLEVEL 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 83 on page 334.

```
Location definitions:
Location Def Mgtype  Ltype  Priority AM Medianames
-------- --- ------  ------ -------- -- -------------------------------------
          N          AUTO     4800   Y
          N          MANUAL   4900   Y
DISTANT   N          STORE     200   Y
LOCAL     N          STORE     300   Y
REMOTE    N          STORE     100   Y
SHELF     N                    5000   Y
```

Figure 83. Sample LISTCONTROL LOCDEF output - No LOCDEFs defined

Task: List the media information defined for your installation.

Command:

```
RMM LISTCONTROL MEDINF
```

Output:

DFSMSrmm displays information such as that shown in Figure 84 on page 335.

```
Media Information :
                                                        Replace Policy
Name     Media type   Recording format Capacity (MB) Perm  Temp  WMC   Age
-------- ------------ ---------------- ------------- ----- ----- ----- -----
MEDINF_A   1 MEDIA1       1 RECTECHA         10000       1     0     0    35
MEDINF_A   0 *            0 *                   0       5     0     0    20
MEDINF_B   2 MEDIA2       1 RECTECHB        150000       1    20    99  9999
MEDINF_B   2 MEDIA2       0 *                   0       1    20    99  9999
MEDINF_C   6 MEDIA6       6 RECTECHC     999999999       1     0     0    35
```

Figure 84. Sample LISTCONTROL MEDINF output

If no media information is defined in EDGRMMxx parmlib, then issuing LISTCONTROL MEDINF produces the output shown in Figure 85 on page 335.

```
NO INSTALLATION DEFINED MEDIA INFORMATION
```

Figure 85. Sample LISTCONTROL MEDINF output - No media information defined

Table 29 on page 335 describes the columns in the LISTCONTROL subcommand output.

Table 29. 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 86 on page 336 shows information that DFSMSrmm displays when there are location definitions in place.

```
 Location definitions:
 Location Def Mgtype  Ltype  Priority AM Medianames
1-------- --- ------  ------ -------- -- -----------------------------------
          N          AUTO     4800   Y
          N          AUTO     4900   Y
 DISTANT  Y  NOBINS  HSTORE   1000   Y  TESTMED1, TESTMED2, TESTMED3
                                        TESTMED4, TESTMED5, 3480
 ILOCBIN  Y  BINS    HSTORE   1000   N  TESTMED1, TESTMED2, TESTMED3
                                        TESTMED4, TESTMED5, 3480
 ILOCNBIN Y  NOBINS  HSTORE   1000   N  TESTMED1, TESTMED2, TESTMED3
                                        TESTMED4, TESTMED5, 3480
 LOCAL    Y  NOBINS  HSTORE   1000   N  TESTMED1, TESTMED2, TESTMED3
                                        TESTMED4, TESTMED5, 3480
 MTL13480 Y          MANUAL    500   Y
 MYLOC    Y          MANUAL   4900   Y
 REMOTE   Y  NOBINS  HSTORE   1000   N  TESTMED1, TESTMED2, TESTMED3
                                        TESTMED4, TESTMED5, 3480
 SHELF    Y                   5000   Y
```

Figure 86. Sample LISTCONTROL LOCDEF output - LOCDEFs defined

Task: List the defaults table entries defined for your installation.

Command:

```
RMM LISTCONTROL DEFTABLE
```

Output:

DFSMSrmm displays information such as that shown in Figure 87 on page 336.

```
DEFTABLE is loaded from PARMLIB member EDGDEFL2, number of entries: 100
DEFTABLE
content:

 ------------------------------------------------------------------------------
DSNAME mask: AAAAAAAA.AAAAAAAA.AAAAAAAA.AAAAAAAA.AAAAAAAA CONTINUE: Y
  JobName mask: AAAAAAAA PGMNAME: AAAAAAAA KeyDate: NOKEYDATE
  RETPD: 93000  RO: N  EDM: N POOL: AAAAAA RM: EXPDT
  (VRSEL) VRSvalue:      VX:
  (EXPDT) LastRef:   WHILECAT: UX    RetainBy: FIRSTFILE
 ------------------------------------------------------------------------------
DSNAME mask: *.FFFFFFFF                                  CONTINUE:
  JobName mask: JOB03    PGMNAME: *         KeyDate: 99051
  RETPD: PERM   RO: N  EDM: N POOL:          RM: VRSEL
  (VRSEL) VRSvalue: VRSVL051  VX: Y
  (EXPDT) LastRef:  91001     WHILECAT: OFF  RetainBy: FIRSTFILE
 ------------------------------------------------------------------------------
```

Figure 87. Sample LISTCONTROL DEFTABLE output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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.
- When the RACF SETROPTS MLNAMES command has been used to activate the name-hiding function, or when COMMANDAUTH(DSN) is in use, to list and search all entries independent of the access granted to the DATASET and TAPEVOL class, you need either CONTROL access to the STGADMIN.EDG.MASTER profile resource or CONTROL access to the STGADMIN.EDG.LIST profile resource.

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.

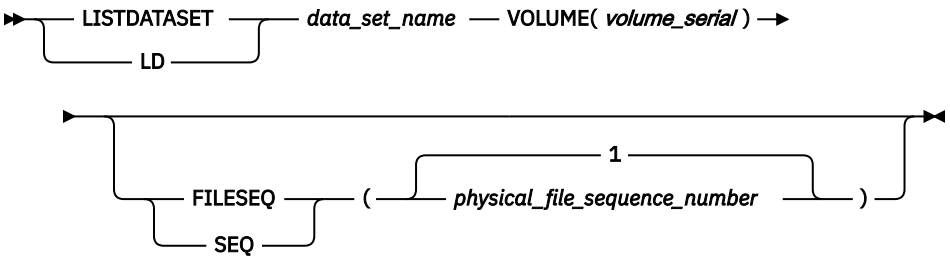
Use the LISTDATASET subcommand to display information about a single data set defined to DFSMSrmm. You must specify the data set name and the volume serial number where the data set resides. If the data set is not the first data set on the volume, you must also specify a sequence number.

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 357 for more information.

Format



Parameters

data_set_name
Specifies the name of the data set about which you want to view information. The name is 1 to 44 characters in length and enclosed in quotes if any special characters are included. If the data set name is not enclosed in quotes, your TSO PROFILE PREFIX value is applied.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

This operand is required and must immediately follow the LISTDATASET subcommand.

FILESEQ(*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.

FILESEQ can be abbreviated as SEQ.

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.

Examples

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 88 on page 339:

```
Data set name = NISHINAL.TEST.ORDER
Volume        = BC0047               Physical file sequence number = 1
Owner         = RMMUSER                        Data set sequence = 1
Create date   = 02/20/2015 Create time = 06:16:06 System ID         = W98MVS1
Expiration date       = 02/20/2016  Expir. time              = 03:00:00
         set by       = OCE_DEF     Original expir.date      =
LASTREF Extra Days    = 14          WHILECATALOG             = OFF
Block size            = 80          Block count              = 3456789012
Data set size(KB)     = 256         FORCEEXPIRE              = NO
Physical size(KB)     = 512         Compression              = 3.00
Percent of volume     = 0           Total block count        = 12345678901234567890
Logical Record Length = 80          Record Format            = FB
Date last written     = 02/20/2015 Date last read            = 02/20/2015
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            =             Closed by Abend          = NO
                                    Deleted                  = NO
VRSEL exclude         = 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   =
BES key index    = 0

Last Change information:
Date             = 02/20/2012  Time = 08:30:59  System      = EZU0000
User change date = 02/20/2012  Time = 08:30:59  User ID     = RMMUSER
```

Figure 88. Sample LISTDATASET output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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.

Format

Parameters

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.

Examples

Task: Request information recorded by DFSMSrmm about the owner whose owner ID is J04735.

Command:

```
RMM LISTOWNER J04735
```

Output: DFSMSrmm displays information such as that shown in Figure 89 on page 340.

```
Owner       = J04735
Last name   = BIGMORE                 First names = Chris J.
Department  = UKIIS Technical Systems MVS
Address     = Mailpoint EW
              North Harbour
              Portsmouth, UK
Telephone:
Internal    = 725-3968                External    = 0705-321212
Electronic mail:
Userid      = BIGMORC                 Node        = CROVM2
Email       = chrisb@storagetek.com

Volumes     = 24

Last Change information:
Date             = 01/05/2011  Time = 08:30:59  System  = EZU0000
User change date = 01/05/2011  Time = 08:30:59  User ID = RMMUSER
```

Figure 89. Sample LISTOWNER output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

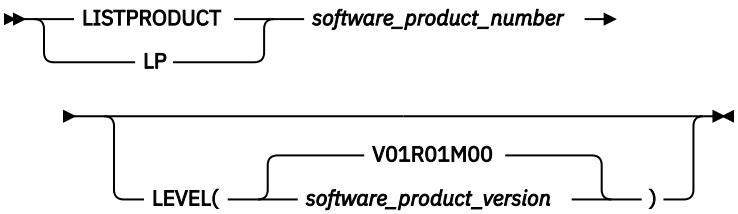
Purpose

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 373 for more information.

Format



Parameters

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 alphanumeric or national characters.

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.

Examples

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 90 on page 342.

```
Product Number = PROD01   Level = V01R01M00  Owner = RMMUSER
Name           = Product One
Description    =

Last Change information:
Date             = 11/05/2011  Time = 08:50:39  System  = EZU0000
User change date = 11/05/2011  Time = 08:31:00  User ID = D008210

Volume    Rack    Feature Code
------   ------   ------------
VOL100   RAC200       1234
VOL101   RAC201       2234
VOL102   RAC202       3234
VOL103   RAC203       4234
VOL104   RAC204       5234
```

Figure 90. Sample LISTPRODUCT output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

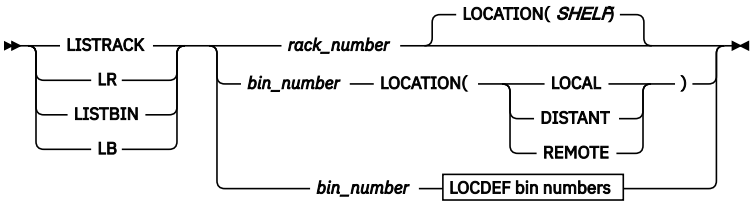
Before you begin: To use the RMM LISTRACK subcommand, you need READ access to the STGADMIN.EDG.MASTER resource profile.

This topic describes the combined description for the LISTRACK subcommand and its alias LISTBIN. See "LISTBIN: Displaying information about a shelf location" on page 325 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 376 and "SEARCHBIN: Creating a list of bin numbers" on page 351 for more information.

Format



LOCDEF bin numbers

Parameters

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.

Examples

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 91 on page 344:

```
Rack/bin number = M00001   Location = MAZBIN     Media name = 3480
Status          = IN USE                        Pool       =
Volume:
  Current       =
  Moving-in     = V10000
  Moving-out    =
  Old           = V10000

Last Change information:
Date             = 01/05/2011  Time = 02:29:53  System  = EZU0000
User change date = 01/05/2011  Time = 02:29:50  User ID = *HKP
```

Figure 91. Sample LISTRACK output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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.
- When the RACF SETROPTS MLNAMES command has been used to activate the name-hiding function, or when COMMANDAUTH(DSN) is in use, to list and search all entries independent of the access granted

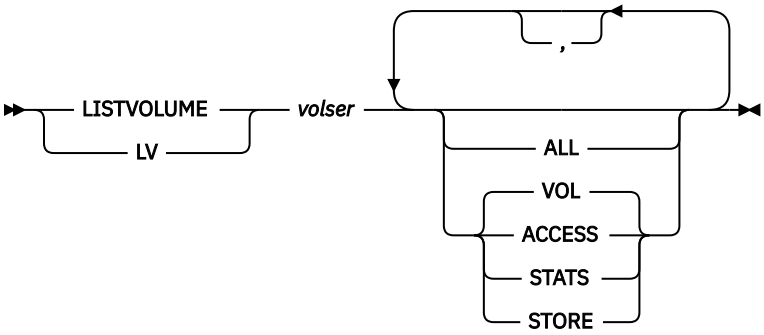
to the DATASET and TAPEVOL class, you need either CONTROL access to the STGADMIN.EDG.MASTER profile resource or CONTROL access to the STGADMIN.EDG.LIST profile resource.

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 380 for more information.

Format



Parameters

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.

Task: Request all the information recorded by DFSMSrmm for the encrypted volume with serial number JJC024.

Command:

```
RMM LISTVOLUME JJC024 ALL
```

Output: DFSMSrmm displays information such as that shown in Figure 92 on page 346:

```
Volume information:
Volume = JJC024   VOL1  =          Rack  = JJC024    Owner   = RMMUSER
  Type = PHYSICAL         Stacked count  = 0         Jobname = D016216J
  Worldwide ID =                                    WORM    = N
Creation:  Date = 02/16/2012  Time = 07:23:23  System ID = W98MVS2
Assign:    Date = 02/16/2012  Time = 07:23:23  System ID = W98MVS2
                                               User ID   =
Expiration date = 09/02/2016             Original      =
Expiration time = 06:57:06   Datasets Kept By Catlg: 0

         set by = OCE_JFCB
Retention date  =            Set retained  = NO
Retention method= VRSEL
         set by = OCE_DEF
      retain by = FIRSTFILE
Data set name = RMMUSER.TEST.CBR
Volume status:             Hold = N   File 1 Data set seq = 1
Status = USER      Availability =                  Label = SL
Current label version =            Required label version =
Media information:
Density = IDRC  Type = EETC      Format = EEFMT4    Compaction = YES
Special attributes   = NONE      Vendor =
Encryption Key Labels:                                          Method:
1=wcc1                                                          LABEL
2=wcc2                                                          LABEL
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 = T,H,IOM,,,SYSPROG
Old loan loc   =
Description    =
Security class =           Description =

Access information:
Owner access = ALTER   Volume access = NONE
VM use = N   MVS use = Y   IRMM use = N
Access list:

Last Change information:
Date             = 02/16/2012  Time = 08:50:39  System  = EZU0000
User change date = 02/16/2012  Time = 08:31:00  User ID = D008210

Statistics:
Number of data sets = 2       Data set recording= ON
Volume usage(MB)= 6           Use count         = 3
Physical(KB)    = 3           Compression       = 3.00
Capacity(MB)    = 59232       Percent full      = 0
Date last read  = 02/16/2012  Date last written = 02/16/2012
Drive last used = 0FA0        Write mount count = 2
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       = ATL15393                                          ATL15393
Type       = AUTO                                              AUTO
Bin number =
Media name =
Priority   =
```

Figure 92. Sample 1: LISTVOLUME output

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 93 on page 347:

```
Volume information:
Volume = 003186   VOL1  = INT001   Rack  = 003186    Owner   = RMMUSER
  Type = PHYSICAL         Stacked count  = 0         Jobname =
  Worldwide ID =                                    WORM    = N
Creation:  Date = 08/28/2011  Time = 04:49:14  System ID = W98MVS2
Assign:    Date =             Time =           System ID = W98MVS2
                                               User ID   =
Expiration date = 09/02/2016         Original      =
Expiration time = 06:57:06   Datasets Kept By Catlg: 0
         set by = OCE_JFCB
Retention date  =            Set retained  = NO
Retention method= VRSEL
         set by = OCE_DEF
      retain by = FIRSTFILE
Data set name = RMMUSER.TEST.CBR
Volume status:             Hold = N   File 1 Data set seq = 1
Status = USER      Availability =                   Label = SL
Current label version =            Required label version =
Media information:
Density = IDRC  Type = EETC      Format = EEFMT2    Compaction = YES
Special attributes   = NONE      Vendor =
Encryption Key Labels:                                          Method:
1=wcc1                                                          LABEL
2=wcc2                                                          LABEL
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 = T,H,IOM,,,SYSPROG
Old loan loc =
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

Last Change information:
Date             = 01/05/2011  Time = 08:50:39  System  = EZU0000
User change date = 01/05/2011  Time = 08:31:00  User ID = D008210
```

Figure 93. Sample 2: 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 94 on page 348:

```
Statistics:
Number of data sets = 32768   Data set recording= ON
Volume usage(MB)= 0           Use count         = 0
Physical(KB)    = 0           Compression       = 0.00
Capacity(MB)    = 0           Percent full      = 0
Date last read  = 11/18/2012  Date last written = 11/18/2012
Drive last used = 0BD9        Write mount count = 0
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 =
Priority   =
```

Figure 94. Sample 3: LISTVOLUME output

Note:

1. The volume usage value in the volume statistics information is calculated from the DCBBLKSIZE multiplied by the number of blocks. For data sets using block sizes larger than 32K prior to implementation of the large block size interface (LBI), 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 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

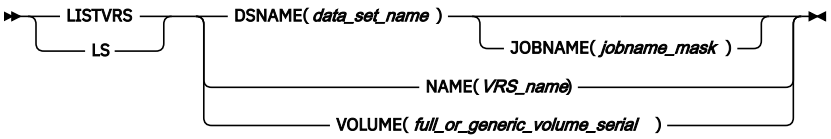
Before you begin: To use the RMM LISTVRS subcommand, you need READ access to the STGADMIN.EDG.VRS resource profile.

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 412 for more information.

Format



Parameters

DSNAME(*data_set_name*)
Specifies the name of the data set for which the vital record specification is defined.

The data set name mask is 1 to 44 characters, enclosed in quotes if any special characters are included. If the data set name mask is not enclosed in quotes, PROFILE PREFIX is applied. This operand is required and must immediately follow the CHANGEVRS subcommand.

DSNAME is mutually exclusive with the NAME and VOLUME operands.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

You can also specify an SMS management class name, a vital record specification management value, or the reserved data set name masks, ABEND, DELETED, or OPEN. The name can be eight alphanumeric characters, beginning with an alphabetic character, and must follow standard z/OS 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.

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 95 on page 350 :

```
RMM ADDVRS DSNAME('A.B') WHILECATALOG
RMM ADDVRS DSNAME('A.B') JOBNAME(BENSJOY) WHILECATALOG
```

Figure 95. 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 96 on page 350:

```
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
Last Reference: Date  = 2010/111   Time  = 12:00:00
Vital Record Specification to be deleted on 31/12/1999

Last Change information:
Date             = 01/05/2011  Time = 08:50:39  System  = EZU0000
User change date = 01/05/2011  Time = 08:31:00  User ID = D008210
```

Figure 96. Sample LISTVRS output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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

Use the SEARCHBIN subcommand 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 30 on page 351 shows the information DFSMSrmm returns for each bin number in the list, in the order it is displayed:

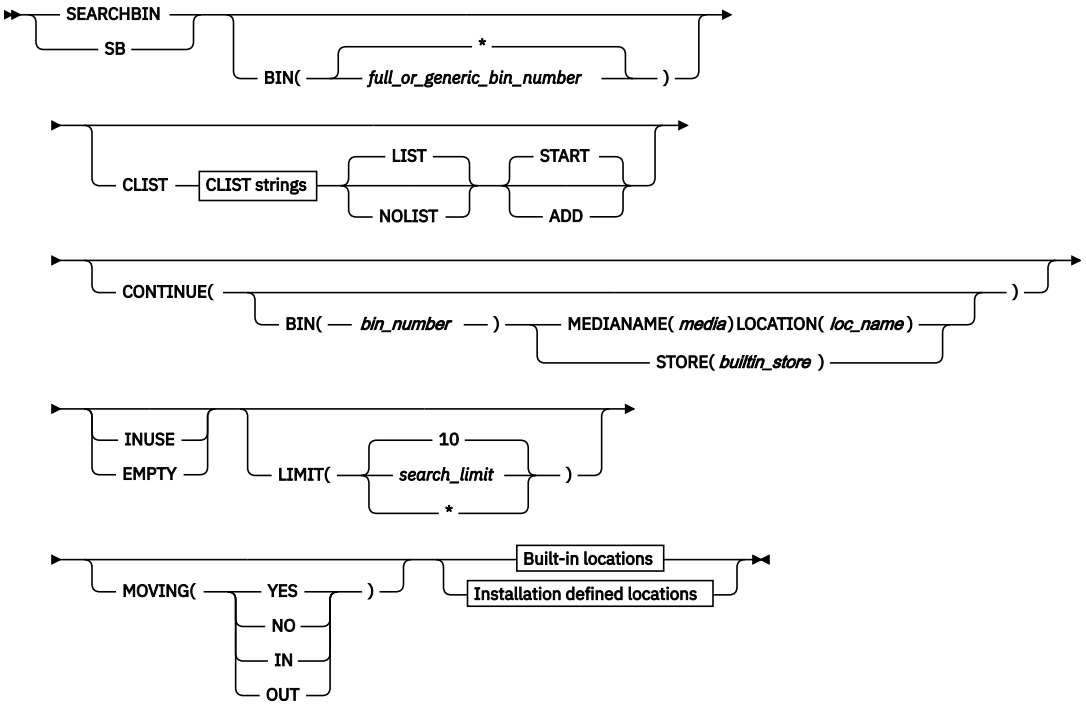
Table 30. Information returned by SEARCHBIN

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

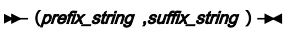
Note: Moving-in volume, moving-out volume, and old volume information is available only if Extended Bin Support is enabled. See the topic Enabling extended bin support in *z/OS DFSMSrmm Implementation and Customization Guide*.

Format

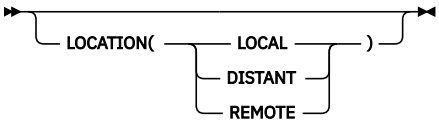
SEARCHBIN syntax diagram



CLIST strings



Built-in locations



Installation defined locations

Parameters

ADD
Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST(''*,'suffix_string*')).

See “Creating CLISTs of executable subcommands” on page 146 for information about the data set used for the CLIST output.

CONTINUE(BIN(*bin_number*) bininfo)
Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a linemode message. When the subcommand is issued from the DFSMSrmm API, the continuation information may be either a linemode message or an SFI or XML attribute.

CONTINUE is an optional operand.

Use the LIMIT operand to control the maximum number of entries to be returned each time you start or continue the search.

To continue a previous search subcommand, the CONTINUE operand value includes these values to identify the current search position:

BIN(*bin_number*)
bin_number is one to six characters enclosed in single quotation marks if it contains any special characters, or blank.

And, either one of the following*bininfo*:

LOCATION(*loc_name*)
loc_name is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

and

MEDIANAME(*media*)
media is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

Or,

STORE(*builtin_store*)
Enter one of the built-in storage location names: LOCAL, DISTANT, REMOTE. Abbreviations are acceptable. For example, L stands for LOCAL.

LOCAL
The local storage location.

DISTANT
The distant storage location.

REMOTE
The remote storage location.

The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, just change the CONTINUE operand value on each additional command required.

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*. For *LOCDEF_location_name*, you can enter any value as no checking is done against the current list of locations defined to DFSMSrmm". 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(*medianame*|*)
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 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 327 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 should neither produce a list nor set REXX variables for resources 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.

START
Specify this operand to request that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

Examples

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 97 on page 355.

```
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 97. 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 98 on page 355.

```
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
000005   LOCAL               EMPTY
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 98. Sample SEARCHBIN output

Task: Create a list of bin numbers using the CONTINUE operand to break down the results of a search into small quantities.

First Command:

```
RMM SEARCHBIN LIMIT(100) CONTINUE
```

First output: DFSMSrmm displays a list such as the one shown in Figure 99 on page 356.

```
Rack/Bin           Media              Current  Moving-in  Moving-out  Old
number   Location  name      Status   volume   volume     volume      volume
-------- --------  --------  -------  -------  ---------  ----------  ------
000001   DISTANT             EMPTY
000002   DISTANT             EMPTY
000003   DISTANT             EMPTY
...
...
...
000003   LOCAL               EMPTY
000004   LOCAL               EMPTY
000005   LOCAL               EMPTY
EDG3203I SEARCH COMPLETE - MORE ENTRIES MAY EXIST
EDG3012I 100        ENTRIES LISTED
EDG3025I BIN('000005')STORE(L)
```

Figure 99. Sample SEARCHBIN output for bins numbers using the CONTINUE operand

Second Command:

```
RMM SEARCHBIN LIMIT(100) CONTINUE(BIN('000005')STORE(L))
```

Second output: DFSMSrmm displays a list such as the one shown in Figure 100 on page 356.

```
Rack/Bin           Media              Current  Moving-in  Moving-out  Old
number   Location  name      Status   volume   volume     volume      volume
-------- --------  --------  -------  -------  ---------  ----------  ------
000006   LOCAL               EMPTY
000007   LOCAL               EMPTY
000008   LOCAL               EMPTY
...
...
...
M00008   MAZBIN    3480      EMPTY
M00009   MAZBIN    3480      EMPTY
M00010   MAZBIN    3480      EMPTY
EDG3203I SEARCH COMPLETE - MORE ENTRIES MAY EXIST
EDG3012I 100        ENTRIES LISTED
EDG3025I BIN('M00010')LOCATION('MAZBIN')MEDIANAME('3480')
```

Figure 100. Sample SEARCHBIN output for bins numbers using the CONTINUE operand

Third Command:

```
RMM SEARCHBIN LIMIT(100) CONTINUE(BIN('M00010')LOCATION('MAZBIN')MEDIANAME('3480'))
```

Third output: DFSMSrmm displays a list such as the one shown in Figure 101 on page 356.

```
Rack/Bin           Media              Current  Moving-in  Moving-out  Old
number   Location  name      Status   volume   volume     volume      volume
-------- --------  --------  -------  -------  ---------  ----------  ------
M00011   MAZBIN    3480      EMPTY
M00012   MAZBIN    3480      EMPTY
M00013   MAZBIN    3480      EMPTY
...
...
...
W00003   WORMSBIN  3490      EMPTY
W00004   WORMSBIN  3490      EMPTY
W00005   WORMSBIN  3490      EMPTY
EDG3012I 65         ENTRIES LISTED
```

Figure 101. Sample SEARCHBIN output for bins numbers using the CONTINUE operand

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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.

  When the RACF SETROPTS MLNAMES command has been used to activate the name-hiding function, or when COMMANDAUTH(DSN) is in use, to list and search all entries independent of the access granted to the DATASET and TAPEVOL class, you need either CONTROL access to the STGADMIN.EDG.MASTER profile resource or CONTROL access to the STGADMIN.EDG.LIST profile resource.

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 31 on page 357 shows the information DFSMSrmm returns for each data set in the order that it is displayed:

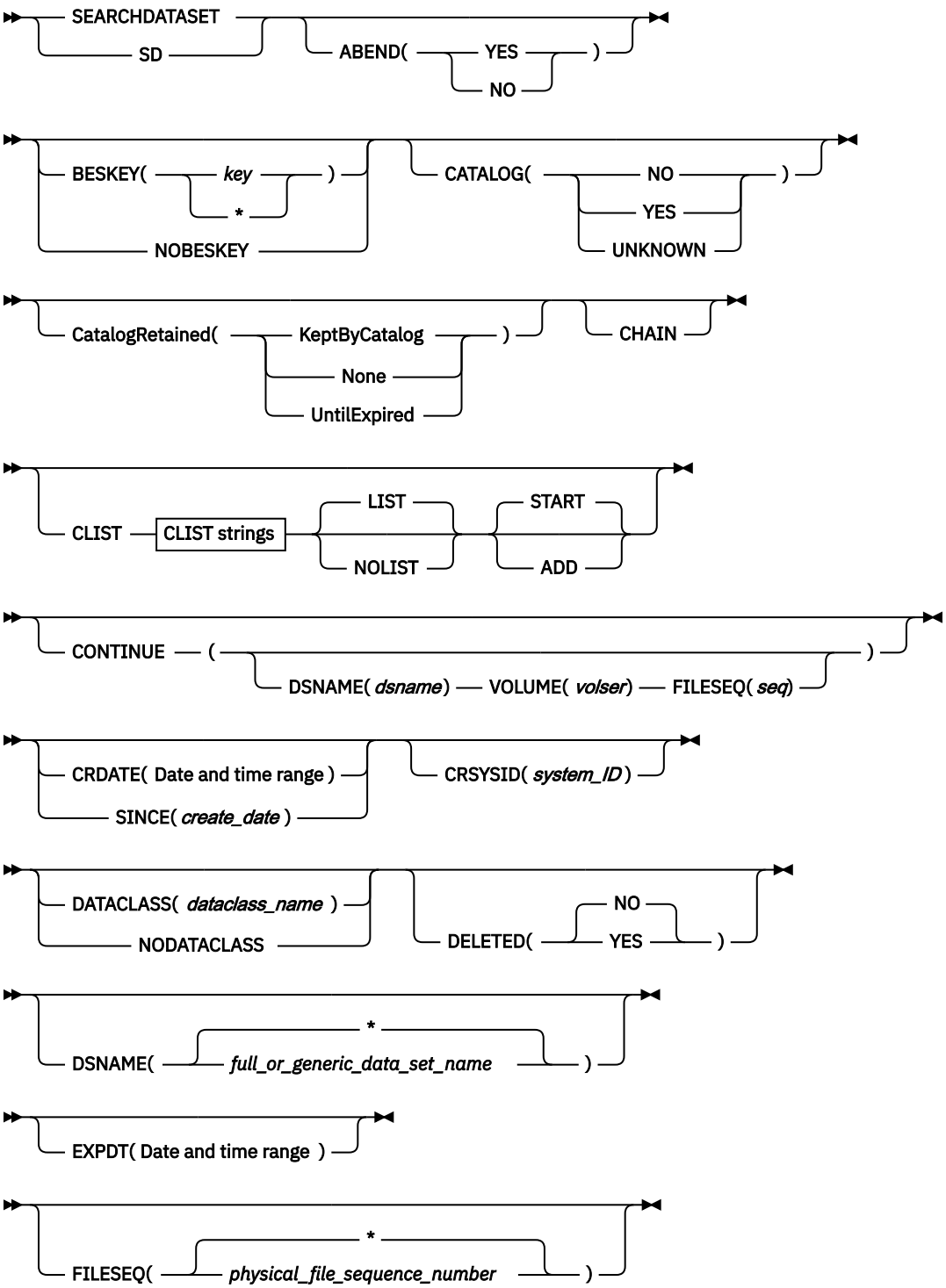
Table 31. Information returned by SEARCHDATASET

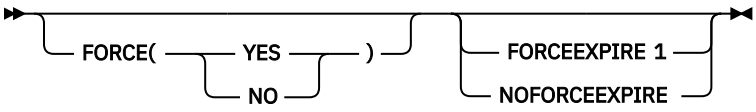
Table field name	Description
Data set name	Data set name
Volume	Volume serial number of the volume on which the data set resides
Owner	Owner ID of the volume owner

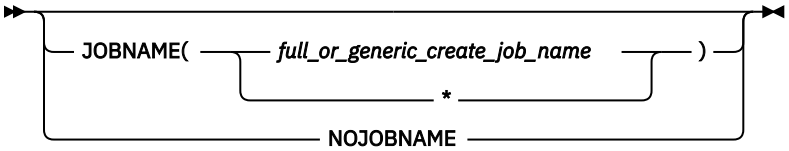
Table 31. Information returned by SEARCHDATASET (continued)

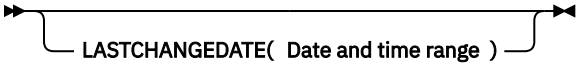
Table field name	Description
Create date	Date the data set was created
Seq	Physical file sequence number

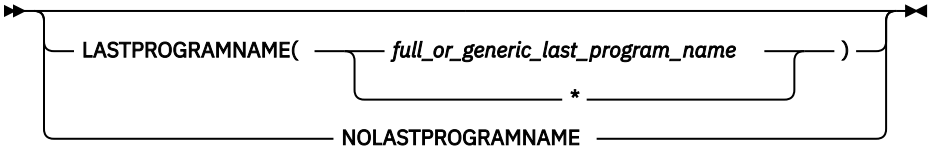
Format

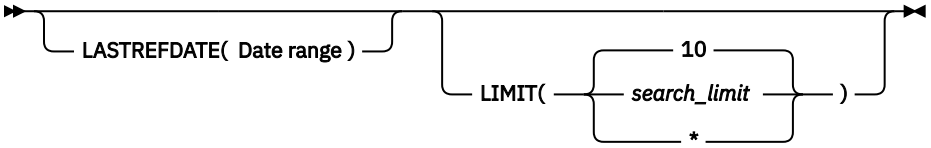


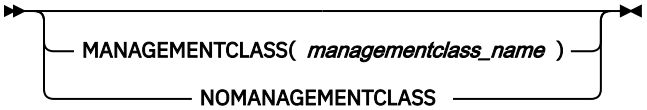

FORCE(
YES
NO
)
FORCEEXPIRE 1
NOFORCEEXPIRE

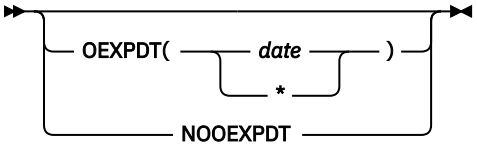

JOBNAME(
full_or_generic_create_job_name
*
)
NOJOBNAME

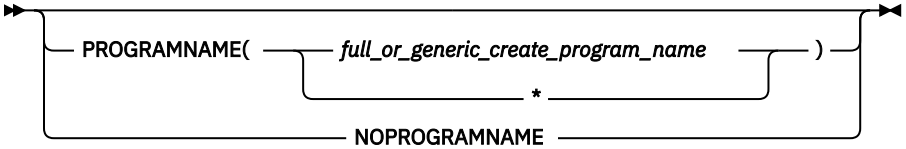

LASTCHANGEDATE(Date and time range)

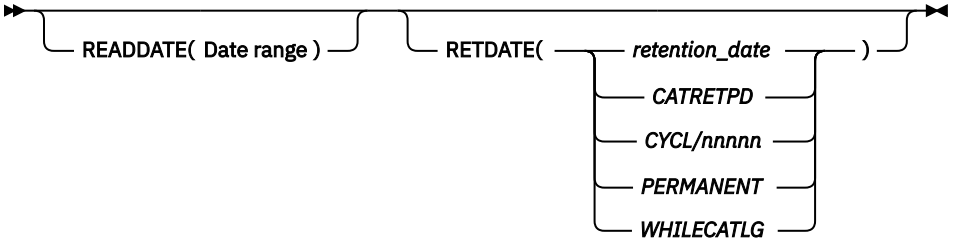

LASTPROGRAMNAME(
full_or_generic_last_program_name
*
)
NOLASTPROGRAMNAME

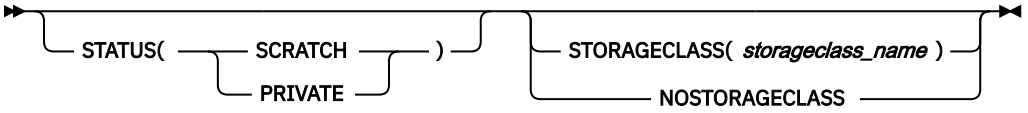

LASTREFDATE(Date range)
LIMIT(
10
search_limit
*
)

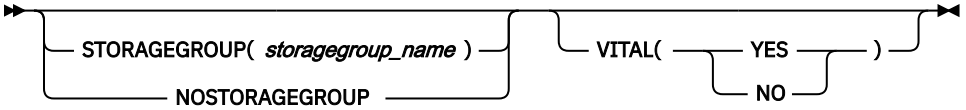

MANAGEMENTCLASS(managementclass_name)
NOMANAGEMENTCLASS

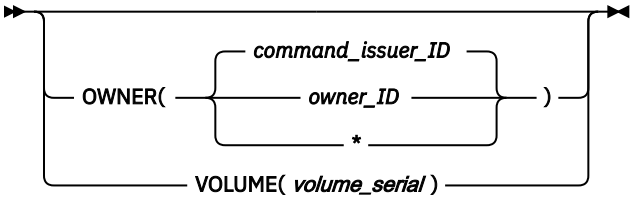

OEXPDT(
date
*
)
NOOEXPDT

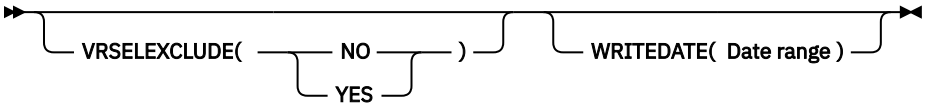

PROGRAMNAME(
full_or_generic_create_program_name
*
)
NOPROGRAMNAME

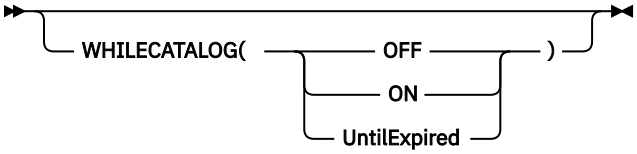

READDATE(Date range)
RETDATE(
retention_date
CATRETPD
CYCL/nnnnn
PERMANENT
WHILECATLG
)


STATUS(
SCRATCH
PRIVATE
)
STORAGECLASS(storageclass_name)
NOSTORAGECLASS

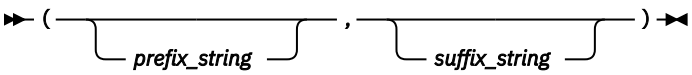

STORAGEGROUP(storagegroup_name)
NOSTORAGEGROUP
VITAL(
YES
NO
)



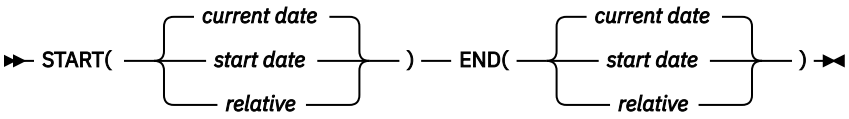




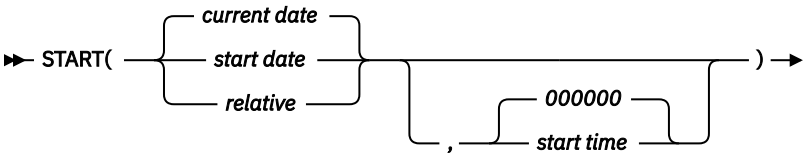
CLIST strings

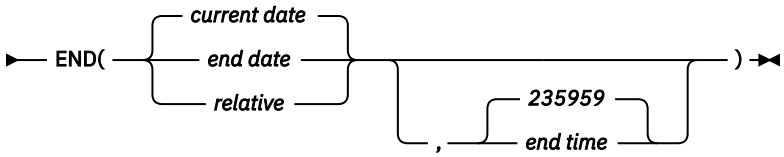


Date range



Date and time range





Note: [1] Can only be specified by a user with CONTROL access to STGADMIN.EDG.MASTER and UPDATE access to STGADMIN.EDG.FORCEEXPIRE security resources. Owner authorization does not apply.

Parameters

ABEND(YES|NO)
Specifies the search to volumes containing data sets that were closed as a result of ABEND processing.

ADD
Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length

can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

BESKEY(*key*)
Specify this operand to search for data sets based on their CA Tape Encryption key index, which is set by the BES subsystem. To search for data sets that have:

- A specific encryption key index value, specify BESKEY(*key*)
- Any nonzero encryption key index value set, specify BESKEY(*)
- No encryption key index value (that is, a key index value of 0), specify NOBESKEY.

CATALOG
Specifies to limit the search to data sets based on catalog status. Specify CATLG(UNKNOWN) to search for data sets that have not yet been cataloged. Specify CATLG(YES) to search for data sets that are currently cataloged. Specify CATLG(NO) to list only data sets that have been uncataloged. CATALOG can be abbreviated as CATLG.

CatalogRetained(KeptByCatalog|None|UntilExpired)
Allows to limit the search to data sets for which expiration depends (or doesn't depend) on their catalog status. WHILECATALOG is an attribute that can only be set for data sets on volumes managed by the EXPDT retention method – CatalogRetained cannot be used to find data sets that are retained by a WHILECATALOG VRS. To find such data sets, SEARCHDATASET RETDATE(WHILECATLG) must be used.

Specify:

CatalogRetained(KeptByCatalog)
to search for data sets that have WHILECATALOG=ON specified and are cataloged. Such data sets show (KeptByCatlg) next to their expiration date to indicate that they are kept past their expiration date as long as they are cataloged.

CatalogRetained(None)
to search for data sets that do not have WHILECATALOG=ON or WHILECATALOG=UntilExpired specified or are uncataloged. Such data sets expire on their expiration date.

Data sets managed by the VRSEL retention method have WHILECATALOG=OFF, and as such are returned if CatalogRetained(None) is specified.

CatalogRetained(UntilExpired)
to search for data sets that have WHILECATALOG=UntilExpired specified and are cataloged. Such data sets show (OrUncatlg) next to their expiration date to indicate that an uncatalog event might decrease the expiration date.

CHAIN
Specifies the search for all physical files in the same multivolume data set. DFSMSrmm returns all the files in the set in volume sequence order starting from the first volume and file in the multivolume data set. You must provide a data set name and a volume serial number of a physical file in the multivolume data set. If you do not specify a file sequence number, DFSMSrmm uses the default file sequence value of 1. The value you specify the number of the physical file on the volume that you specify for the search. The output might consist of files with numbers which are different from this number; a multivolume data set can start on any physical file on a volume, but on each subsequent volume the file sequence number is 1.

When you use the CHAIN operand, DFSMSrmm ignores all other operands you specify except for the ADD, CLIST, LIST, NOLIST, START operands.

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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST('','*suffix_string*')).

See "Creating CLISTs of executable subcommands" on page 146 for information about the data set used for the CLIST output.

CONTINUE(DSNAME(*dsname*)VOLUME(*volser*)FILESEQ(*seq*))
Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a line mode message. When the subcommand is issued from the DFSMSrmm API, the continuation information can be either a line mode message or an SFI or XML attribute.

CONTINUE is an optional operand.

Use the LIMIT operand to control the maximum number of entries to be returned each time that you start or continue the search.

To continue a previous search subcommand, the CONTINUE operand value includes all of these values to identify the current search position:

DSNAME(*dsname*)
dsname is one to 44 characters enclosed, in single quotation marks if it contains any special characters, or blank. The default is "*" (an asterisk).

VOLUME(*volser*)
volser is one to six characters, enclosed in single quotation marks if it contains any special characters, or blank.

FILESEQ(*seq*)
seq is a number within the value range 0 to 65535.

The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, just change the CONTINUE operand value on each additional command required.

CRDATE
Lists the data sets whose creation date and time match the specified date and time criteria. CRDATE is mutually exclusive with the SINCE operand. CRDATE can be specified as any of the following:

CRDATE
Only data sets whose creation date is the current date are listed

CRDATE(START(*start_date,start_time*))
Only data sets whose creation date is on or after the specified start date and (if specified) start time are listed, where *start_date* is either an absolute date or relative date and *start_time* specifies the start time criteria.

CRDATE(END(*end_date,end_time*))
Only data sets whose creation date is on or before the specified end date and (if specified) end time are listed, where *end_date* is either an absolute date or relative date and *end_time* specifies the start time criteria. Note that because START defaults to the current date, the specified end date must be equal or greater than the current date when START is omitted.

CRDATE(START(*start_date,start_time*)END(*end_date,end_time*))
Only data sets whose creation date and creation time are within the range delimited by the specified start and end dates and times are listed, where both *start_date* and *end_date* are either an absolute date or relative date. The specified end date must be equal to or greater than the specified start date.

Each of the *start_date* and *end_date* values can be absolute or relative dates.

Absolute dates are specified as either yyyy/ddd or yyddd format. For example, January 3, 2015 can be specified as 2015/003 or 15003.

Relative dates are specified as a number of days, months, or years beforeto the current date.

-0
specifies the current day, current month, current year.

-n
specifies that the date is *n* days before the current date

-nM
specifies that the date is *n* months before the current month and the current day in the month is as the current date.

-nY
specifies that the date is *n* years before the current year and the current day in the year is as the current date.

The value range for *n* is 0 to 99999, with a required leading dash ('-') and an optional suffix of M or Y.

Start and end times are specified in the *hhmmss* format, where:

hh
is hours

mm
is minutes

ss
is seconds

and assumes a 24 hour clock. For example, nine PM in the evening is 210000. You can specify a time in the range from 000000 to 235959.

The default start_time is 000000. The default end_time is 235959.

Note: If *start_time* is specified, then *start_date* must also be specified and if *end_time* is specified, then *end_date* must also be specified.

Examples: To list data sets whose creation date is:

Any time today
specify: SD CRDATE

Today between six AM and eleven AM in the morning
specify: SD CRDATE(START(-0,060000) END(-0,110000))

Three days ago between eleven AM in the morning and 5:30 PM
specify: SD CRDATE(START(-3,110000) END(-3,173000))

Before January 1, 2000
specify: SD CRDATE(START(0000/001) END(1999/365))

On or after eight AM in the morning, January 2, 2015
Specify: SD CRDATE(START(2015/002,080000))

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.

DATACLASS(*dataclass_name*)
Specifies to limit the search to data sets with the specified data class name. A data class name is one-to-eight alphanumeric, national, or special characters. DATACLASS is mutually exclusive with NODATACLASS.

DELETED(NO | YES)
Specifies to limit the search to data sets based on deleted status. Specify:

DELETED(YES)
to search for data sets that are deleted.

DELETED(NO)
to list only data sets that are not deleted.

The default value is NO.

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. DFSMSrmm does not check data set names for valid characters. Any string of up to 44 characters is accepted, except those that start with a blank or x'00'. Data set names without quotations have your TSO PROFILE PREFIX value applied. Any data set name containing generic characters is validated against the basic mask rules. However, if the mask rules are not met, the data set name is treated as non-generic.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these will not match to data sets with all uppercase characters.

In addition to normal data set naming conventions, you can use these 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.

Period (.)
A leading or trailing period is not allowed. Consecutive periods are also not allowed.

Double asterisk (**)
Cannot be specified within a data set name qualifier.

Cannot be specified before a qualifier with a single asterisk. For example, specifying DSNAME(**.*) is not a valid data set name mask.

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.

The default value is *.

EXPDT(*date_and_time_range*)
Lists the data sets whose current expiration date matches the specified date criteria. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHDATASET CRDATE operand for a description of how to use the START and END suboperands and examples of their use.

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 *.

FORCE
Limits the search based on whether the force flag of a data set is set. If the force flag is set, it means that information about the data set has been changed by a CHANGEDATASET subcommand with the FORCE operand and the requested change was made only because FORCE was specified. Specify FORCE(YES) to list only those data sets whose force flag is set on. Specify FORCE(NO) to list only those data sets whose force flag is not set.

There is no default value for FORCE. If FORCE is not specified, data sets are listed without regard to how their force flag is set.

FORCEEXPIRE
Specify the FORCEEXPIRE operand to select data sets with the FORCEEXPIRE attribute. FORCEEXPIRE is used to manually block access to a data set.

When specifying the FORCEEXPIRE keyword, DFSMSrmm always checks for the STGADMIN.EDG.FORCEEXPIRE and STGADMIN.EDG.MASTER security resources, independent of the other parameters specified.

UPDATE access to STGADMIN.EDG.FORCEEXPIRE and CONTROL access to STGADMIN.EDG.MASTER is needed to complete the search.

FORCEEXPIRE and NOFORCEEXPIRE are mutually exclusive.

JOBNAME(*create_jobname*|*)

Specifies the job name that created the data set. A job name is one-to-eight alphanumeric characters or $, #, or @. You can use a generic job name. Use % in your generic job name mask to match any one character and * to match any character string in the jobn ame.

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.

If you have data sets with job names that include symbols other than alphanumeric characters, $, #, or @, use a generic job name to find them.

LASTCHANGEDATE(*date_and_time_range*)
Lists the data sets whose last changed date and time matches the specified date criteria. The date and time criteria can be specified using the START and END suboperands. See the description of the SEARCHDATASET CRDATE operand for a description of how to use the date and time range and examples of its use.

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.

LASTREFDATE(*date_range*)
Lists the data sets based on both the last read date and last write date, using the most recent of both dates. The most recent of the two values within the date range for a data set to be selected. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHDATASET CRDATE operand for a description of how to use the START and END suboperands and examples of their use.

LIMIT(*search_limit*)
Specifies the number of entries that DFSMSrmm lists. The maximum allowable decimal value is 9999. More than 9999 entries can be displayed by specifying an asterisk to request a list of all entries matching your search criteria. LIMIT is mutually exclusive with END.

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.

MANAGEMENTCLASS(*managementclass_name*)
Specifies to limit the search to data sets with the specified management class name. A management class name is one-to-eight alphanumeric, national, or special characters. MANAGEMENTCLASS is mutually exclusive with NOMANAGEMENTCLASS.

NODATACLASS
Specifies to limit the search to data sets that do not have a data class. NODATACLASS is mutually exclusive with DATACLASS.

NOJOBNAME
Specifies data sets that do not have a job name. NOJOBNAME is mutually exclusive with JOBNAME.

NOLIST
Specifies that DFSMSrmm should neither produce a list nor set REXX variables for resources 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.

NOMANAGEMENTCLASS
Specifies to limit the search to data sets that do not have a management class. NOMANAGEMENTCLASS is mutually exclusive with MANAGEMENTCLASS.

NOPROGRAMNAME
Specifies to limit the search to data sets that do not have a creating program name. NOPROGRAMNAME is mutually exclusive with PROGRAMNAME.

NOOEXPDT
Specifies to limit the search to data sets that do not have an original expiration date. NOOEXPDT is mutually exclusive with OEXPDT.

NOFORCEEXPIRE
Specify the NOFORCEEXPIRE operand to select data sets without the FORCEEXPIRE attribute. FORCEEXPIRE is used to manually block access to a data set.

When specifying the NOFORCEEXPIRE keyword, DFSMSrmm always checks for the STGADMIN.EDG.FORCEEXPIRE and STGADMIN.EDG.MASTER security resources, independent of the other parameters specified.

UPDATE access to STGADMIN.EDG.FORCEEXPIRE and CONTROL access to STGADMIN.EDG.MASTER is needed to complete the search.

FORCEEXPIRE and NOFORCEEXPIRE are mutually exclusive.

NOSTORAGECLASS
Specifies to limit the search to data sets that do not have a storage class. NOSTORAGECLASS is mutually exclusive with STORAGECLASS.

NOSTORAGEGROUP
Specifies to limit the search to data sets that do not have a storage group. NOSTORAGEGROUP is mutually exclusive with STORAGEGROUP.

OEXPDT(*date*)
Lists the data sets whose original expiration date matches the specified date.

If you use an *, DFSMSrmm returns data set information for all data sets that have any original expiration date. OEXPDT is mutually exclusive with NOOEXPDT.

OWNER(*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 data set 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.

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.

READDATE(*date_range*)
Lists the data sets whose last read date matches the specified date criteria. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHDATASET CRDATE operand for a description of how to use the START and END suboperands and examples of their use.

RETDATE(*retention_date*)
RETDATE specifies that DFSMSrmm lists only data sets that expire up to and including the specified date. Data sets on scratch volumes are excluded. You can specify a specific date as RETDATE(*retention_date*). You can also specify the DFSMSrmm special date formats; CATRETPD, PERMANENT, WHILECATLG, or CYCL/*nnnnn*, where "*nnnnn*" is five numeric digits. When you specify one of the special dates, DFSMSrmm lists only those data sets that are VRS retained with that special retention date. When you specify the special cycles format date, CYCL/*nnnnn*, DFSMSrmm lists data sets that are VRS retained and have a cycles retention date and the same number or fewer cycles. For example, RETDATE(CYCL/00255) searches for all data sets with a retention date set to CYCL/00255 or lower, such as CYCL/00001. For data sets retained by a VRS DFSMSrmm uses the retention date. For data sets not retained by a vital record specification, DFSMSrmm uses the expiration date for the search. To obtain a list of data sets 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 2012/001. 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 12001. 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 RETDATE operand, DFSMSrmm searches all data sets, regardless of their retention date. See “Defining retention policies for data sets and volumes” on page 77 for information about how DFSMSrmm calculates retention dates.

SINCE(*create_date*)
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 2012/001. 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 12001.

If you do not specify the SINCE parameter, DFSMSrmm considers all data sets regardless of the date when they were created.

SINCE is mutually exclusive with the CRDATE operand.

START
Requests that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

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.

STORAGECLASS(*storageclass_name*)
Specifies to limit the search to data sets with the specified storage class name. A storage class name is one-to-eight alphanumeric, national, or special characters. STORAGECLASS is mutually exclusive with NOSTORAGECLASS.

STORAGEGROUP(*storagegroup_name*)
Specifies to limit the search to data sets with the specified storage group name. A storage group name is one-to-eight alphanumeric, national, or special characters. STORAGEGROUP is mutually exclusive with NOSTORAGEGROUP.

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 to limit 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.

VRSELEXCLUDE(YES | NO)
Specifies to limit the search to data sets excluded (or not excluded) from VRSEL processing. Specify NO to search for data sets that are not excluded from VRSEL processing. Specify YES to search for data sets that are excluded from VRSEL processing. VRSELEXCLUDE can be abbreviated as VX.

WHILECATALOG(Off|On|UntilExpired)
Allows to limit the search to data sets that have WHILECATALOG set to a particular value. WHILECATALOG is an attribute that can only be set for data sets on volumes managed by the EXPDT retention method.

Specify:

WHILECATALOG(Off)
to search for data sets that have WHILECATALOG=OFF specified. The expiration of such data sets does not depend on their catalog status. Note that data sets that are retained by a WHILECATALOG VRS also have WHILECATALOG set to OFF.

WHILECATALOG(On)
to search for data sets that have WHILECATALOG=ON specified. Such data sets are kept past their expiration date if they are cataloged.

WHILECATALOG(UntilExpired)
to search for data sets that have WHILECATALOG=UntilExpired specified. Such data sets expire after they are uncataloged, but no later than their expiration date.

WRITEDATE(*date_range*)
Lists the data sets whose last write date matches the specified date criteria. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHDATASET CRDATE operand for a description of how to use the START and END suboperands and examples of their use.

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 102 on page 369:

```
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 102. 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 103 on page 369:

```
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 103. 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 104 on page 369:

```
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 104. 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 105 on page 370:

```
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 105. Sample SEARCHDATASET output

DFSMSrmm also creates a CLIST data set containing the records shown in Figure 106 on page 370:

```
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 106. Sample CLIST data set

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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.

SEARCHOWNER: Searching owner information

Purpose

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

Use the SEARCHOWNER subcommand to create a list of owners defined to DFSMSrmm. You can restrict how many owners DFSMSrmm displays by specifying the LIMIT or END operand. DFSMSrmm searches

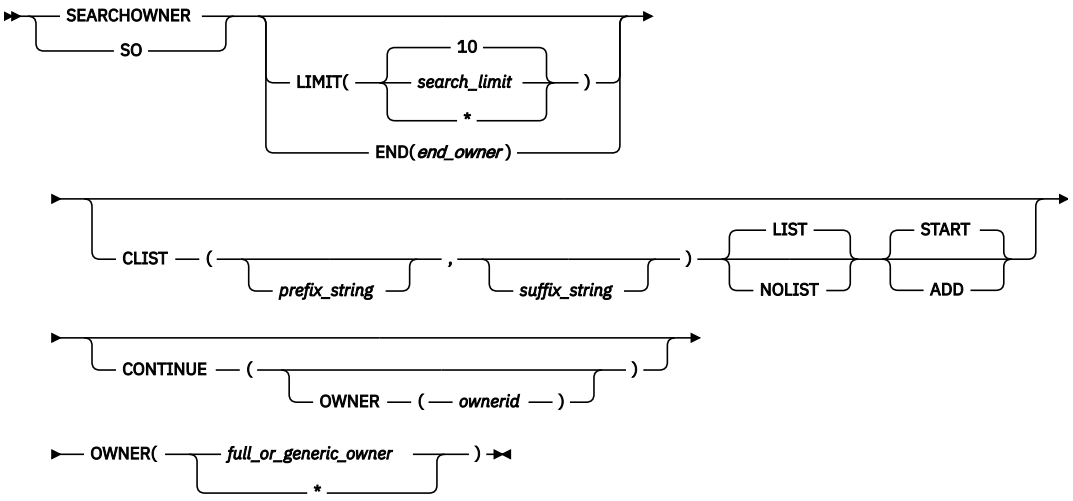
until it reaches your limit or end point, or until it lists all owners that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten.

Table 32 on page 371 shows the information DFSMSrmm returns for each owner in the list, in the order it is displayed:

Table 32. Information returned by the RMM SEARCHOWNER subcommand

Table field name	Description
Owner	Owner's ID
Last Name	Owner's last name
First Names	Owner's first names
Telephone	Owner's internal telephone number
Volumes	Number of volumes owned

Format



Parameters

ADD

Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

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 owners you do not want in the list. Then you can run the CLIST at your convenience.

DFSMSrmm returns the owner serial number for each record if you do not specify (*prefix_string* and *suffix_string*). When the owner 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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST('','*suffix_string*')).

See “Creating CLISTs of executable subcommands” on page 146 for information about the data set used for the CLIST output.

CONTINUE(OWNER(*ownerid*))
Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a line mode message. When the subcommand is issued from the DFSMSrmm API, the continuation information may be either a line mode message or an SFI or XML attribute.

CONTINUE is an optional operand.

Use the LIMIT operand to control the maximum number of entries to be returned each time you start or continue the search.

To continue a previous search subcommand, the CONTINUE operand value includes this value to identify the current search position:

OWNER(*ownerid*)
ownerid is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, just change the CONTINUE operand value on each additional command required.

END(*end_owner*)
Specify END as an alternative to the LIMIT operand to enable you to specify both the starting and ending point of the owner search.

END is mutually exclusive with LIMIT.

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.

LIMIT is mutually exclusive with END.

The default value is 10.

LIST
Specifies that DFSMSrmm produce a list when the CLIST operand is used.

LIST is mutually exclusive with the NOLIST operand. LIST is the default.

NOLIST
Specifies that DFSMSrmm should neither produce a list nor set REXX variables for resources 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(*full_or_generic_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.

START
Specify this operand to request that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

Task: Create a list of all owners defined to DFSMSrmm.

Command:

```
RMM SEARCHOWNER OWNER(*) LIMIT(*)
```

Output: DFSMSrmm displays a list such as the one shown in Figure 107 on page 373.

```
Owner    Last Name            First Names          Internal Volumes
 -------- -------------------- -------------------- -------- ----------
 WOODMW   Wood                 Mike                 664358   50
```

Figure 107. Sample SEARCHOWNER output listing

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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.

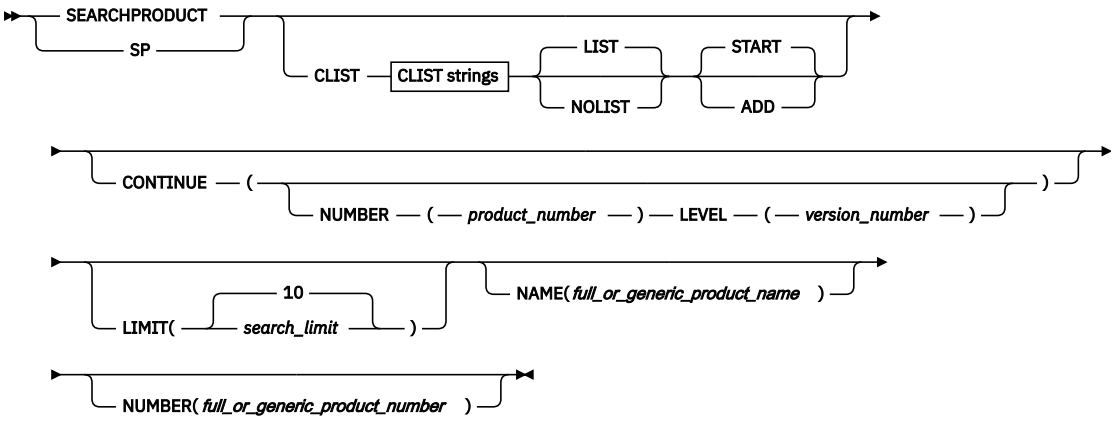
You can restrict how many software products DFSMSrmm displays by specifying the LIMIT operand. DFSMSrmm searches until it reaches your limit or until it lists all software products that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten products.

Table 33 on page 374 shows the information DFSMSrmm returns for each software product in the list, in the order it is displayed:

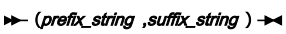
Table 33. Information returned by SEARCHPRODUCT

Table field name	Description
Number	Software product number
Level	Software product level
Product Name	Software product name
Feature Code	Software product feature code
Vols	Number of volumes associated with the software product
Volume	Volume serial number of the first volume where the software product resides

Format



CLIST strings



Parameters

ADD

Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST(''*suffix_string*')).

See "Creating CLISTs of executable subcommands" on page 146 for information about the data set used for the CLIST output.

CONTINUE(NUMBER(*product_number*)LEVEL(*version_number*))
Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a line mode message. When the subcommand is issued from the DFSMSrmm API, the continuation information may be either a line mode message or an SFI or XML attribute.

CONTINUE is an optional operand.

Use the LIMIT operand to control the maximum number of entries to be returned each time you start or continue the search.

To continue a previous search subcommand, the CONTINUE operand value includes all of these values to identify the current search position:

NUMBER(*product_number*)
product_number is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

LEVEL(*version_number*)
version_number is one to nine characters enclosed in single quotation marks if it contains any special characters or blank.

The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, just change the CONTINUE operand value on each additional command required.

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 should neither produce a list nor set REXX variables for resources 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.

START
Specify this operand to request that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

Examples

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 108 on page 376.

```
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 108. Sample SEARCHPRODUCT output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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.

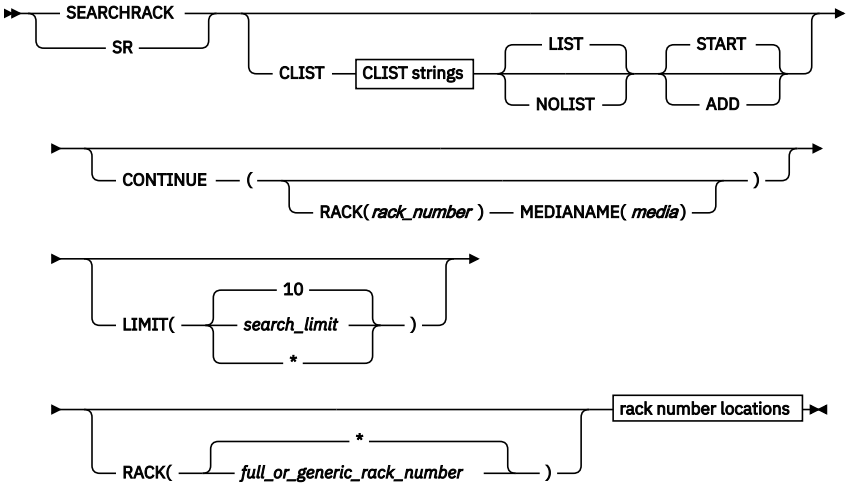
You can restrict the number of rack numbers DFSMSrmm lists by specifying the LIMIT operand. DFSMSrmm searches until it reaches your limit or until it lists all shelf locations that match your search criteria. If you do not specify a search limit, DFSMSrmm lists a maximum of ten shelf locations.

Table 34 on page 377 shows the information DFSMSrmm returns for each rack number in the list, in the order it is displayed:

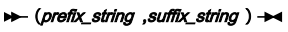
Table 34. Information returned by SEARCHRACK

Table field name	Description
Rack	Rack number
Medianame	Type of volume
Volume	Volume serial number
Status	Status of the shelf location (one of EMPTY, INUSE, or SCRATCH)
Location	Location where the volume resides

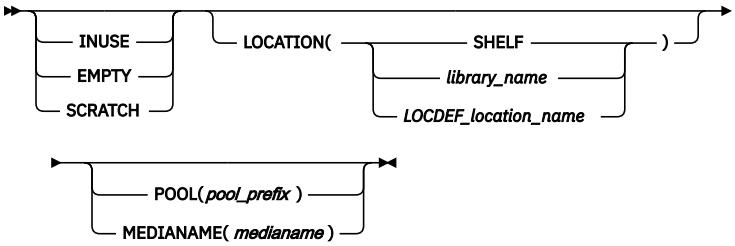
Format



CLIST strings



rack number locations



Parameters

ADD

Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length

can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST(''*,'suffix_string*')).

See “Creating CLISTs of executable subcommands” on page 146 for information about the data set used for the CLIST output.

CONTINUE(RACK(*rack_number*)MEDIANAME(*media*))
Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a line mode message. When the subcommand is issued from the DFSMSrmm API, the continuation information may be either a line mode message or an SFI or XML attribute.

CONTINUE is an optional operand.

Use the LIMIT operand to control the maximum number of entries to be returned each time you start or continue the search.

To continue a previous search subcommand, the CONTINUE operand value includes all of these values to identify the current search position:

RACK(*rack_number*)
rack_number is one to six characters enclosed in single quotation marks if it contains any special characters, or blank.

MEDIANAME(*media*)
media is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, just change the CONTINUE operand value on each additional command required.

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_name)
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(*medianame*)
Specifies a list that is limited to shelf locations containing volumes with the same media name. Media names are defined by your installation and 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 327 for more information.

NOLIST
Specifies that DFSMSrmm should neither produce a list nor set REXX variables for resources 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 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.

START
Specify this operand to request that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

Examples

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 109 on page 380.

```
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 109. Sample SEARCHRACK output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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.

SEARCHVOLUME: Creating a list of volumes

Purpose

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.
- When the RACF SETROPTS MLNAMES command has been used to activate the name-hiding function, or when COMMANDAUTH(DSN) is in use, to list and search all entries independent of the access granted

to the DATASET and TAPEVOL class, you need either CONTROL access to the STGADMIN.EDG.MASTER profile resource or CONTROL access to the STGADMIN.EDG.LIST profile resource.

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. For example, you can display lists of volumes based on ownership, assigned date, status, movement, action, pool, media name, and many others.

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.

Use the TYPE(LOGICAL) and CLIST operands to create an output file that can help you to prepare an import list.

Table 35 on page 381 shows the information DFSMSrmm returns for each volume in the list, in the order it is displayed:

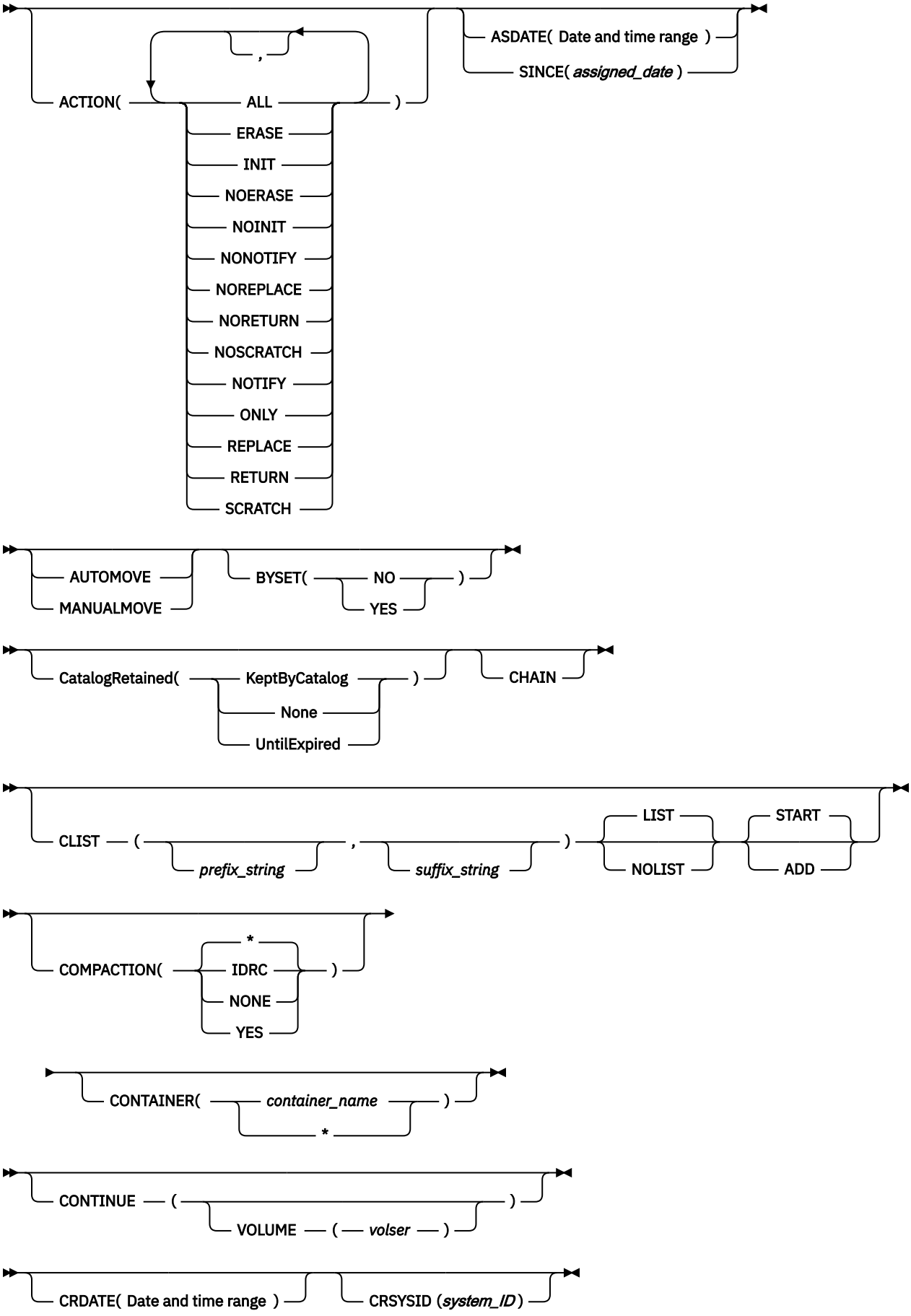
Table 35. Information returned by the RMM SEARCHVOLUME subcommand

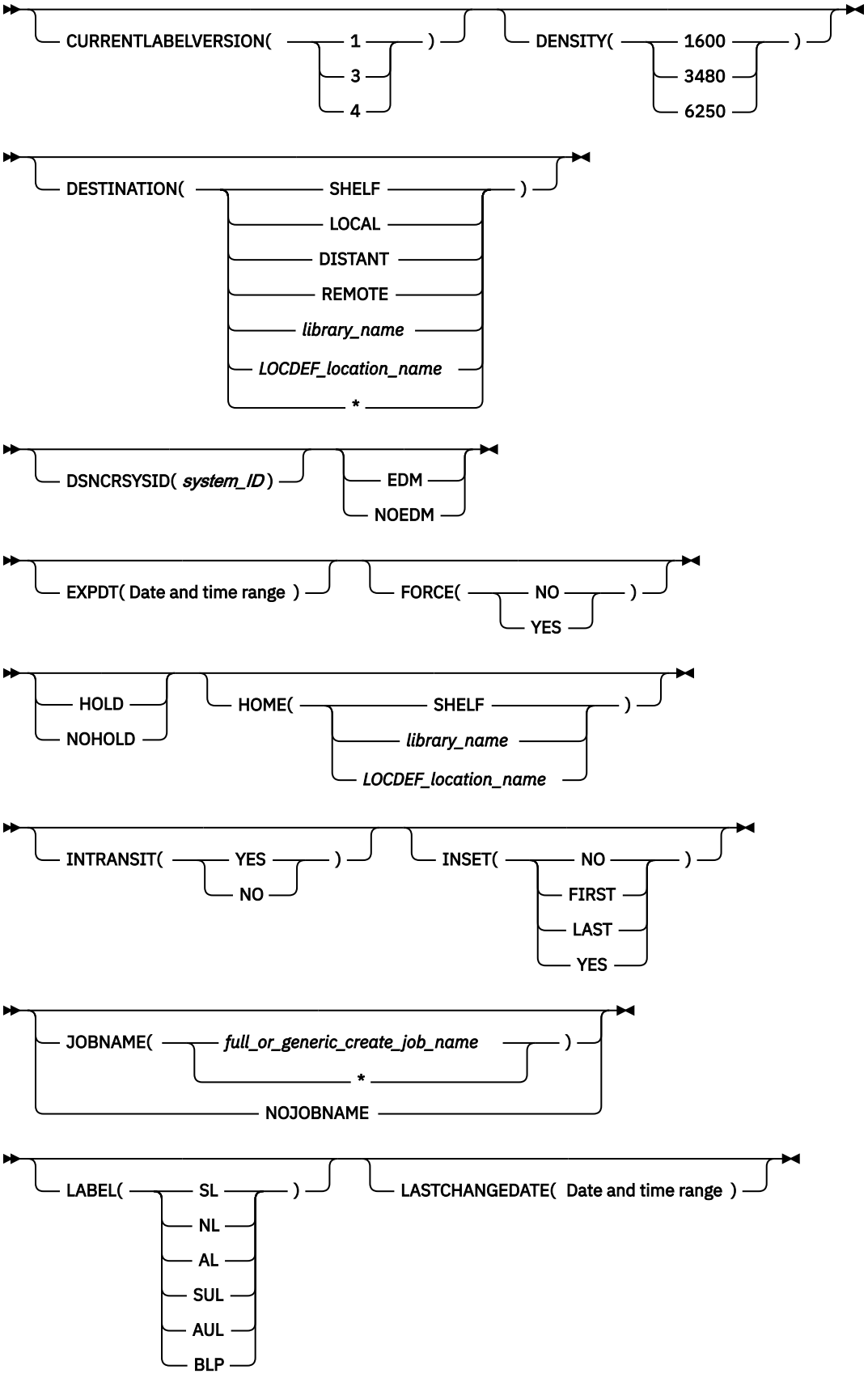
Table field name	**Description**
Volume	Volume serial number
Owner	Owner ID of the volume owner
Rack	Rack number
Assigned date	• Date the volume is assigned to a user • Date the volume is in MASTER status when a non-specific volume is requested in a batch job • Date the volume is returned to scratch 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

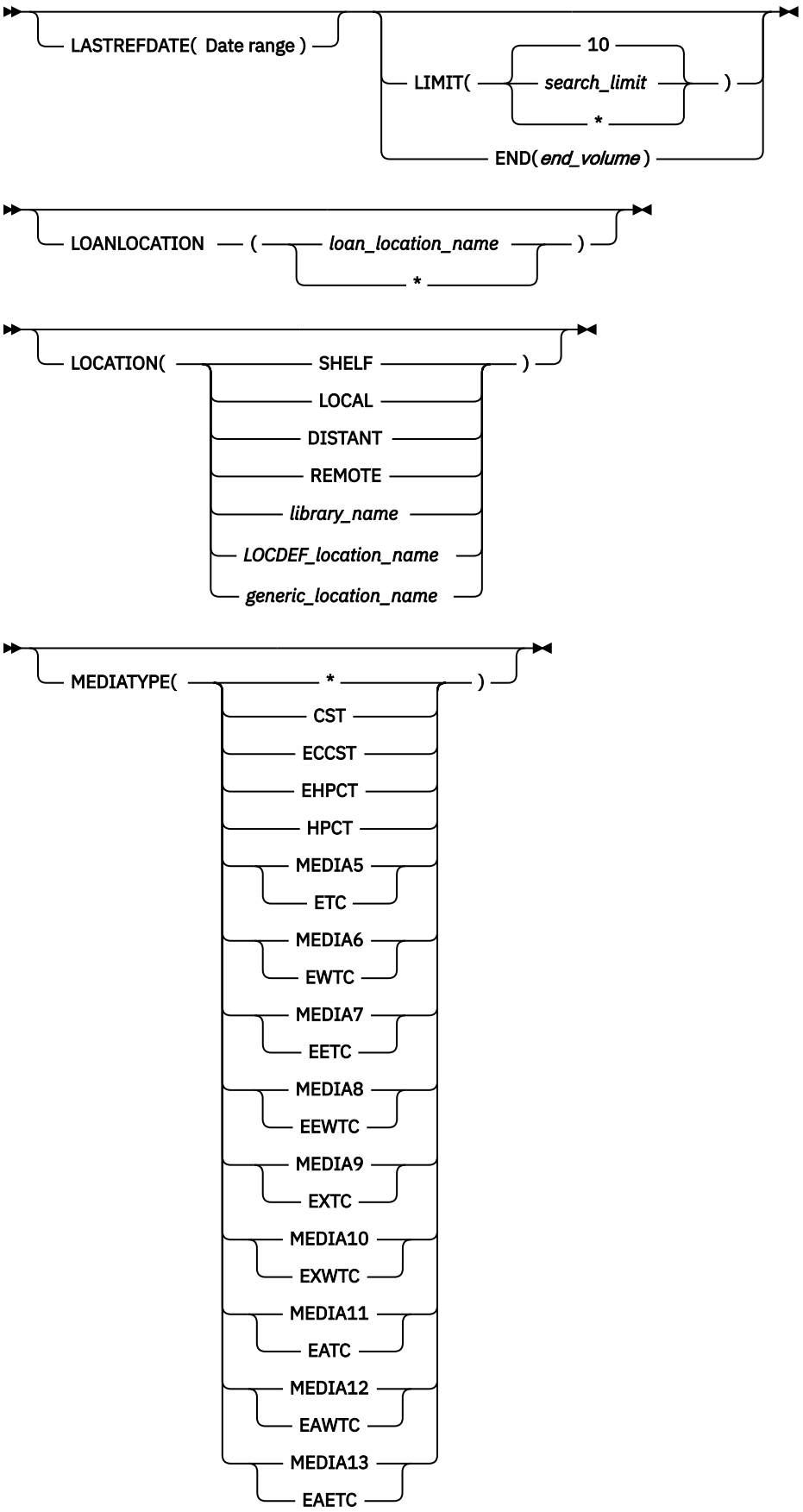
Table 35. Information returned by the RMM SEARCHVOLUME subcommand (continued)

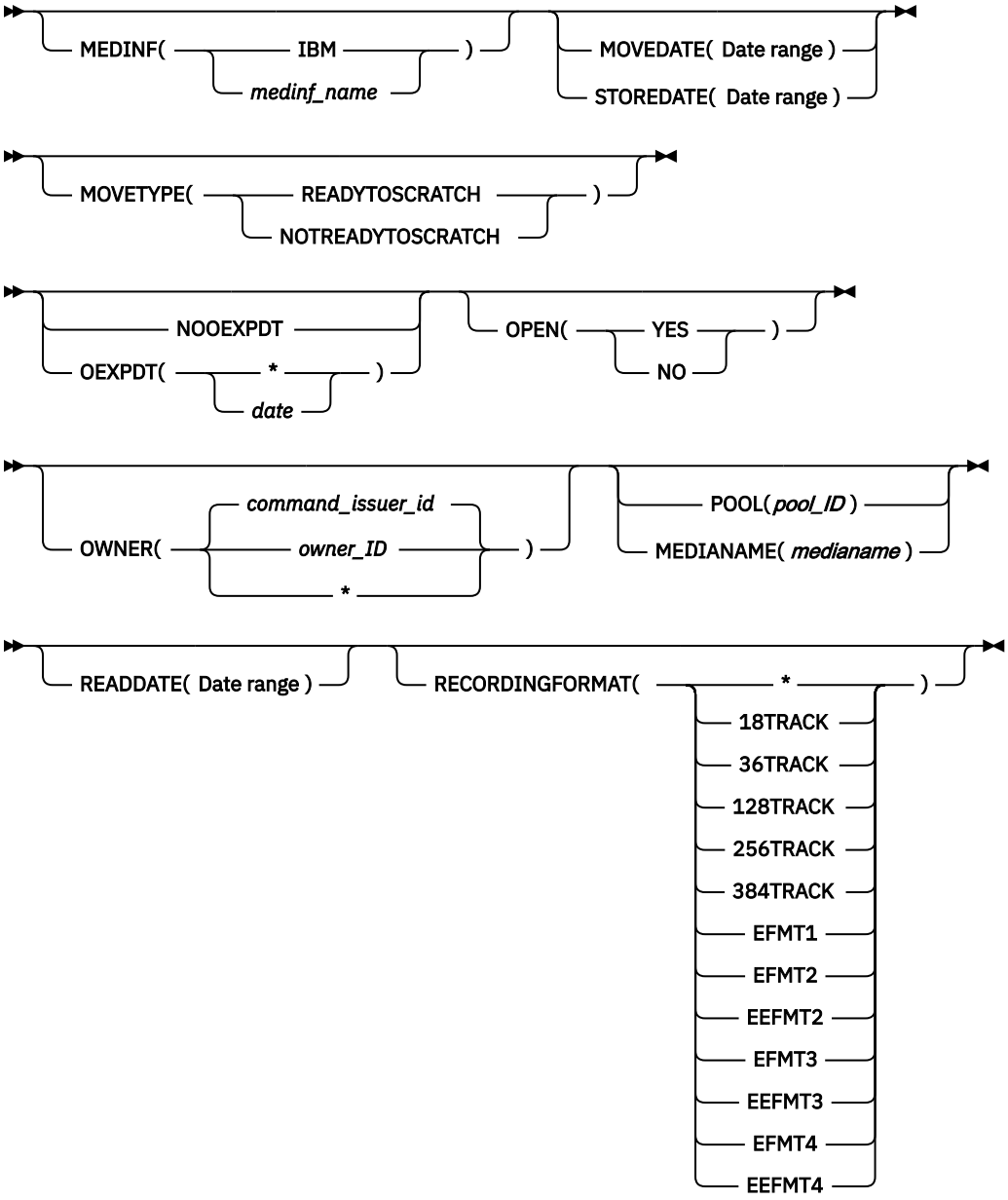
Table field name	Description
St	Volume's status and availability, abbreviated as follows: **E** Scratch volume awaiting entry into a system-managed tape library **I** Scratch volume awaiting initialization **L** Volume is on loan **M** Master volume **O** Volume is open for output **R** Volume is pending release **S** Scratch volume **U** User volume **V** Vital record
Act	Type of release action specified for the volume: **E** Erase **I** Initialize **N** Notify owner **O** Return to owner **R** Replace **S** Return to scratch
Dest.	Location where the volume is moving to. (One of SHELF, HOME, LOCAL, DISTANT, REMOTE, a *library_name*, or a *LOCDEF_location_name*)

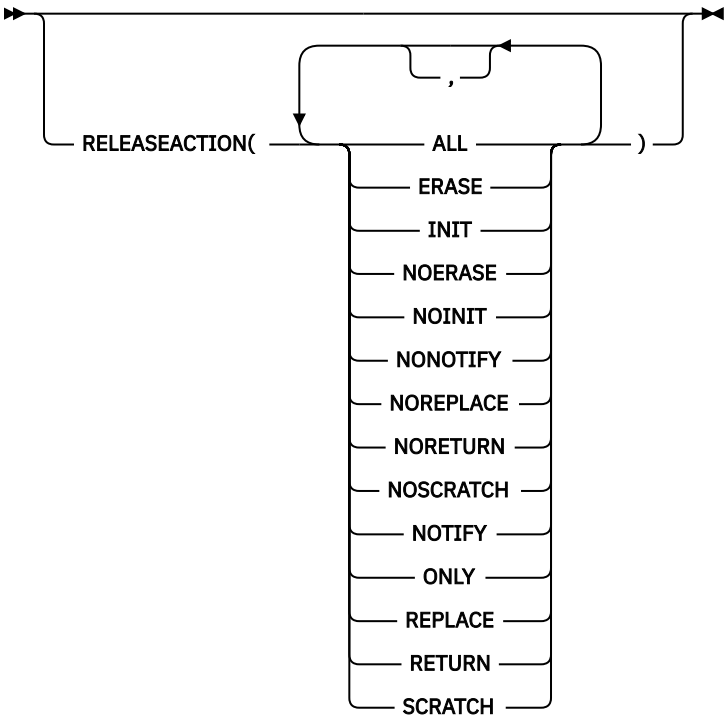
Format

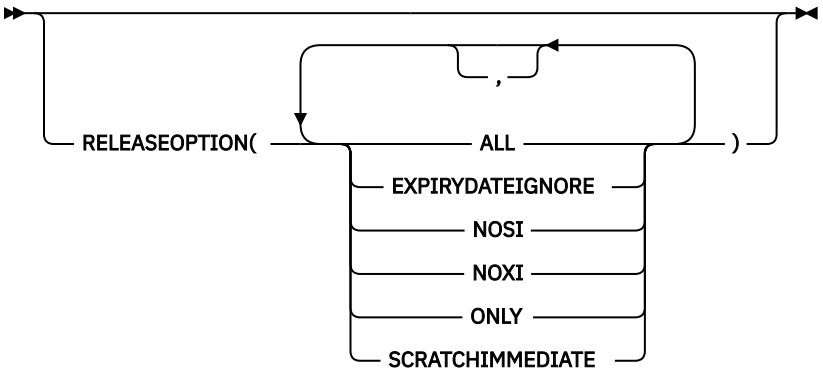

ACTION(
,
ALL
ERASE
INIT
NOERASE
NOINIT
NONOTIFY
NOREPLACE
NORETURN
NOSCRATCH
NOTIFY
ONLY
REPLACE
RETURN
SCRATCH
)
ASDATE(Date and time range)
SINCE(assigned_date)
AUTOMOVE
MANUALMOVE
BYSET(
NO
YES
)
CatalogRetained(
KeptByCatalog
None
UntilExpired
)
CHAIN
CLIST
(
prefix_string
,
suffix_string
)
LIST
NOLIST
START
ADD
COMPACTION(
*
IDRC
NONE
YES
)
CONTAINER(
container_name
*
)
CONTINUE
(
VOLUME
(
volser
)
)
CRDATE(Date and time range)
CRSYSID (system_ID)

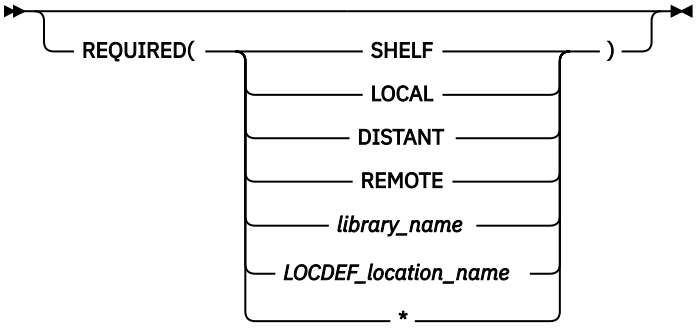

CURRENTLABELVERSION(1 3 4)
DENSITY(1600 3480 6250)
DESTINATION(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name *)
DSNCRSYSID(system_ID) EDM NOEDM
EXPDT(Date and time range)
FORCE(NO YES)
HOLD NOHOLD
HOME(SHELF library_name LOCDEF_location_name)
INTRANSIT(YES NO)
INSET(NO FIRST LAST YES)
JOBNAME(full_or_generic_create_job_name *)
NOJOBNAME
LABEL(SL NL AL SUL AUL BLP)
LASTCHANGEDATE(Date and time range)

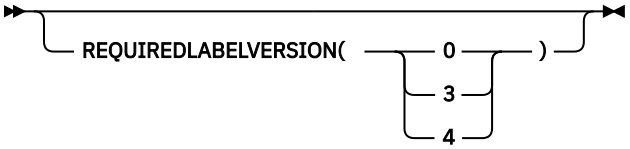

LASTREFDATE(Date range)
LIMIT(10 search_limit *)
END(end_volume)
LOANLOCATION (loan_location_name *)
LOCATION(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name generic_location_name)
MEDIATYPE(* CST ECCST EHPCT HPCT MEDIA5 ETC MEDIA6 EWTC MEDIA7 EETC MEDIA8 EEWTC MEDIA9 EXTC MEDIA10 EXWTC MEDIA11 EATC MEDIA12 EAWTC MEDIA13 EAETC)

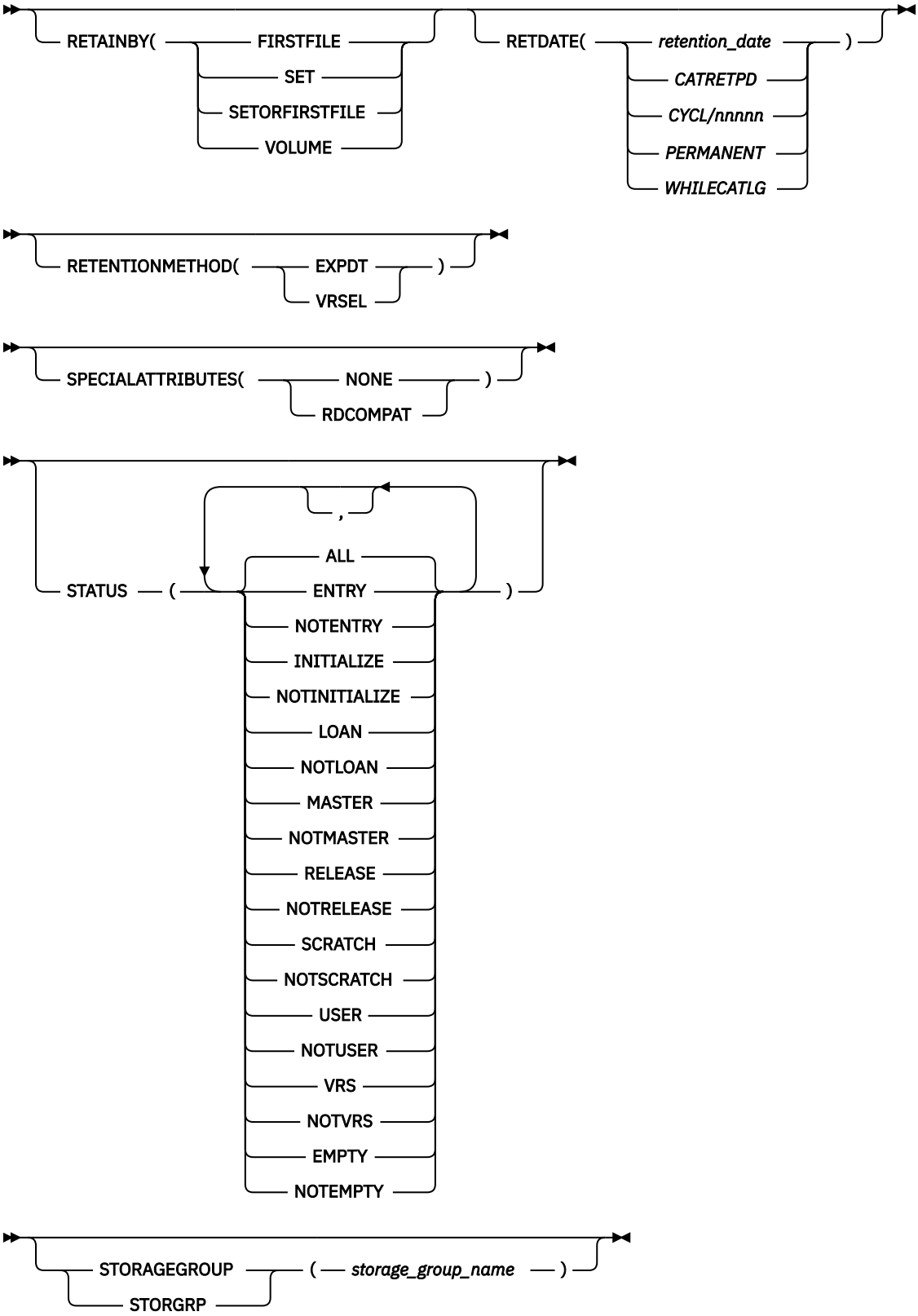

MEDINF(
IBM
medinf_name
)
MOVEDATE(Date range)
STOREDATE(Date range)
MOVETYPE(
READYTOSCRATCH
NOTREADYTOSCRATCH
)
NOOEXPDT
OEXPDT(
*
date
)
OPEN(
YES
NO
)
OWNER(
command_issuer_id
owner_ID
*
)
POOL(pool_ID)
MEDIANAME(medianame)
READDATE(Date range)
RECORDINGFORMAT(
*
18TRACK
36TRACK
128TRACK
256TRACK
384TRACK
EFMT1
EFMT2
EEFMT2
EFMT3
EEFMT3
EFMT4
EEFMT4
)

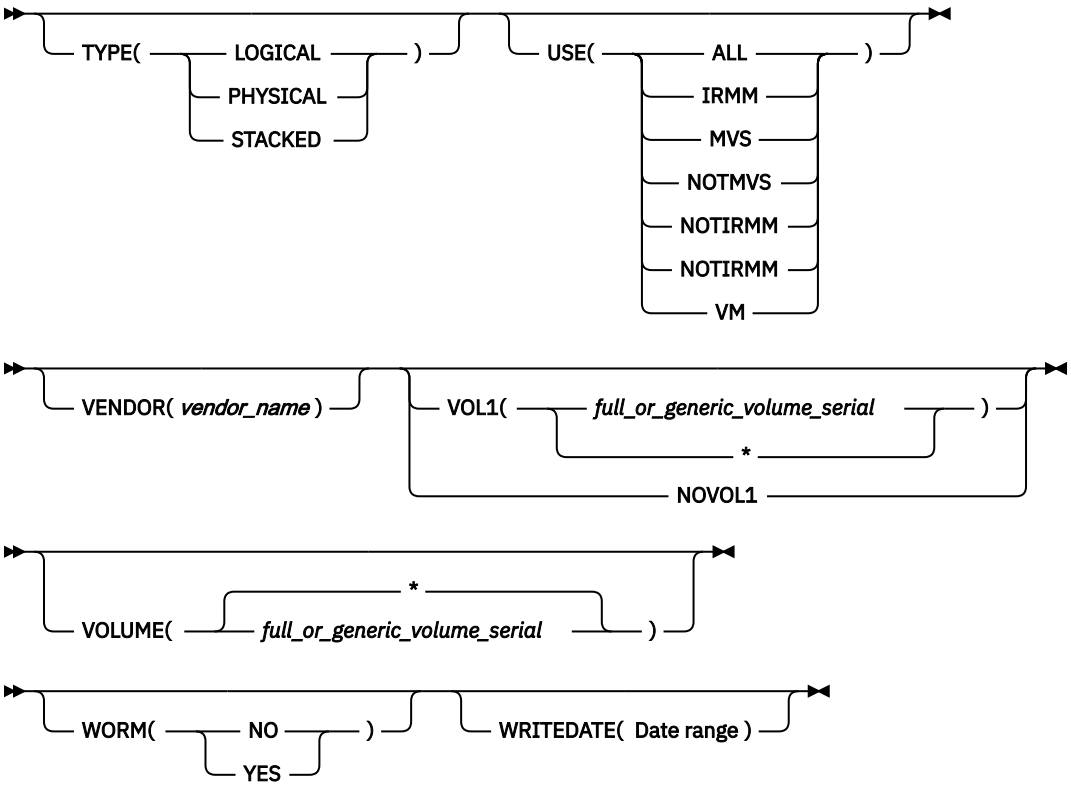

RELEASEACTION(
,
ALL
)
ERASE
INIT
NOERASE
NOINIT
NONOTIFY
NOREPLACE
NORETURN
NOSCRATCH
NOTIFY
ONLY
REPLACE
RETURN
SCRATCH


RELEASEOPTION(
,
ALL
)
EXPIRYDATEIGNORE
NOSI
NOXI
ONLY
SCRATCHIMMEDIATE

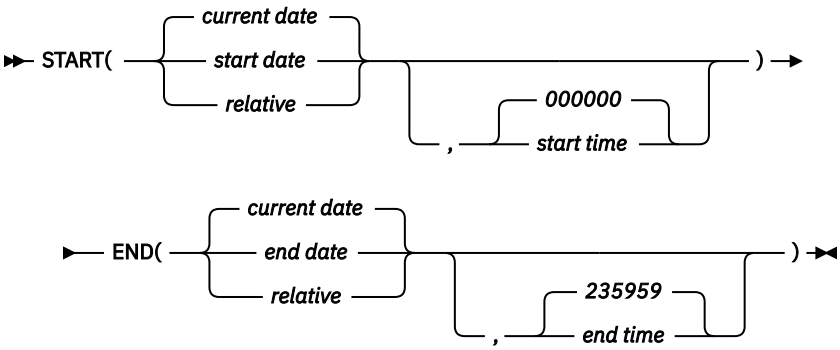

REQUIRED(
SHELF
)
LOCAL
DISTANT
REMOTE
library_name
LOCDEF_location_name
*


REQUIREDLABELVERSION(
0
)
3
4


RETAINBY(
FIRSTFILE
SET
SETORFIRSTFILE
VOLUME
RETDATE(
retention_date
CATRETPD
CYCL/nnnnn
PERMANENT
WHILECATLG
)
RETENTIONMETHOD(
EXPDT
VRSEL
)
SPECIALATTRIBUTES(
NONE
RDCOMPAT
)
STATUS
(
,
ALL
ENTRY
NOTENTRY
INITIALIZE
NOTINITIALIZE
LOAN
NOTLOAN
MASTER
NOTMASTER
RELEASE
NOTRELEASE
SCRATCH
NOTSCRATCH
USER
NOTUSER
VRS
NOTVRS
EMPTY
NOTEMPTY
)
STORAGEGROUP
STORGRP
(
storage_group_name
)



Date and time range



Parameters

ABEND(YES | NO)
Specifies to limit the search to volumes containing a data set that was closed as a result of ABEND processing.

ACTION
Specifies one or more actions DFSMSrmm uses as a search criteria. DFSMSrmm lists only volumes with the indicated pending release action. Use the **ACTION(*pending_actions*)** operand to search for volumes that have the specified release action set. Volumes are returned if any of the values you specify are set in the volume.

Specify one or more of the following values for *pending_actions*, 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.

NOTIFY
To list only volumes for which owners notified.

REPLACE
To list only volumes that replaced by new volumes and returned to the scratch pool.

RETURN
To list only volumes that should be returned to their owner.

SCRATCH
To list all volumes to be returned to scratch status.

NOERASE
Used in combination with ALL or other operands to exclude volumes that require erasing.

NOINIT
Used in combination with ALL or other operands to exclude volumes that require initialization.

NONOTIFY
Used in combination with ALL or other operands to exclude volumes for which owners notified.

NOREPLACE
Used in combination with ALL or other operands to exclude volumes that replaced by new volumes and returned to the scratch pool.

NORETURN
Used in combination with ALL or other operands to exclude volumes that should be returned to their owner.

NOSCRATCH
Used in combination with ALL or other operands to exclude volumes to be returned to scratch status.

ONLY
Used in combination with other operands to limit the list to a specific combination. For example, ACTION(RETURN,NOTIFY,ONLY) will list only those volumes for which **both** the RETURN and NOTIFY pending release action apply. If ONLY were omitted, then DFSMSrmm would list all the volumes for which **either** pending release action applied.

When you specify ONLY with no other operand value, DFSMSrmm selects volumes that have no pending release action set. This is the same as specifying NOERASE, NOINIT, NONOTIFY, NOREPLACE, NORETURN, NOSCRATCH.

ADD
Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

ASDATE
Lists the volumes whose assigned date and time matches the specified date criteria. ASDATE is mutually exclusive with the SINCE operand. ASDATE can be specified as any of the following:

ASDATE
Only volumes whose assigned date is the current date are listed

ASDATE(START(*start_date*))
Only volumes whose assigned date is on or after the specified start date are listed, where *start_date* is either an absolute date or relative date.

ASDATE(END(*end_date*))
Only volumes whose assigned date is on or before the specified end date are listed, where *end_date* is either an absolute date or relative date. Note that because START defaults to the current date, the specified end date equal to or greater than the current date when START is omitted.

ASDATE(START(*start_date*)END(*end_date*))
Only volumes whose assigned date is within the range delimited by the specified start and end dates are listed, where both *start_date* and *end_date* are either an absolute date or relative date. The specified end date equal to or greater than the specified start date.

Each of the *start_date* and *end_date* values can be absolute or relative dates.

Absolute dates are specified as either yyyy/ddd or yyddd format. For example, January 3, 2011 can be specified as 2011/003 or 11003.

Relative dates are specified as a number of days, months, or years prior to the current date.

-0
specifies the current day, current month, current year.

-n
specifies that the date is *n* days before the current date

-nM
specifies that the date is *n* months before the current month and the current day in the month is as the current date.

-nY
specifies that the date is *n* years before the current year and the current day in the year is as the current date.

The value range for *n* is 0 to 99999, with a required leading '-' and an optional suffix of M or Y.

Examples: To list volumes whose assigned date is:

Today
specify: SV ASDATE

Three days ago
specify: SV ASDATE(START(-3) END(-3))

Before January 1, 2000
specify: SV ASDATE(START(0000/001) END(1999/365))

On or after January 2, 2005
Specify: SV ASDATE(START(2005/002))

AUTOMOVE
Specifies a list of volumes that have a move mode of AUTOMOVE which indicates that DFSMSrmm automatic movement processing is in effect.

BYSET(YES | NO)
Specifies a list of volumes based on the setting of the "set retained" attribute in the record, which is set when a volume is retained only because it is part of a multivolume set and one or more other volumes in the set are retained. Refer to the RETAINBY parmlib option. Specify BYSET(YES) to list volumes that have the "set retained" attribute set. Specify BYSET(NO) to list volumes that do **not** have the "set retained" attribute set.

CatalogRetained(KeptByCatalog|No|UntilExpired)
Allows to limit the search for volumes for which expiration depends (or doesn't depend) on data set catalog status. `CatalogRetained` cannot be used to find volumes that are retained by a WHILECATALOG VRS because WHILECATALOG is an attribute that can only be set for data sets on volumes managed by the EXPDT retention method. To find such volumes, SEARCHVOLUME RETDATE(WHILECATLG) must be used.

Specify:

CatalogRetained(KeptByCatalog)
to search for volumes that show (KeptByCatlg) next to their expiration date to indicate that they are kept past their expiration date because of a data set that is cataloged.

CatalogRetained(No)
to search for volumes for which expiration is not influenced by the WHILECATALOG attribute. They do not show (KeptByCatlg) or (OrUncatlg) next to their expiration date, and expire on the expiration date.

Volumes managed by the VRSEL retention method always have WHILECATALOG=OFF for all of their data sets, and as such are returned if `CatalogRetained(No)` is specified even if their retention date is “WHILECATLG”.

CatalogRetained(UntilExpired)
to search for volumes that show (OrUncatlg) next to their expiration date to indicate that an uncatalog event might decrease the expiration date due to data sets that have WHILECATALOG=UntilExpired set.

CHAIN
Specifies all volumes in the same multivolume data set. You must specify a specific volume serial number for any volume in the data set. DFSMSrmm retrieves the volume information from the control data set. DFSMSrmm uses the previous and next volume chain to return all volumes in the data set in volume sequence order, starting from the first volume in the data set.

When you use the CHAIN operand, DFSMSrmm ignores all other operands you specify except for the ADD, CLIST, LIST, NOLIST, START operands.

This operand is optional and has no default.

CLIST(*prefix_string,suffix_string*)
Specifies a CLIST to create a data set of executable commands or to prepare an import list. 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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST('','*suffix_string*')).

When you specify the TYPE(LOGICAL) operand and CLIST, DFSMSrmm returns more information in the output file if obtained logical volume resides on a stacked volume. In such a case, 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.

See “Creating CLISTs of executable subcommands” on page 146 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 these:

*****
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 being searched is compacted.

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 volume that is defined to DFSMSrmm. The value can be any alphanumeric or special characters up to 6 characters in length.

CONTINUE(VOLUME(*volser*))
Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a line mode message. When the subcommand is issued from the DFSMSrmm API, the continuation information can be either a line mode message or an SFI or XML attribute.

CONTINUE is an optional operand.

Use the LIMIT operand to control the maximum number of entries to be returned each time you start or continue the search.

To continue a previous search subcommand, the CONTINUE operand value includes this value to identify the current search position:

VOLUME(*volser*)
volser is one to six characters enclosed in single quotation marks if it contains any special characters, or blank.

The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, just change the CONTINUE operand value on each additional command required.

SKIP(*skip-number*))
Allows you to display search results after a specified number of matching entries are skipped.

skip-number is any number from 0 to 4,294,967,295. The default value is 0.

SKIP is mutually exclusive with CONTINUE.

Specifying SKIP causes a count of the number of entries to be displayed in the command output. This count includes those entries that were skipped over due to the SKIP(n) parameter.

CRDATE
Lists the volumes whose creation date and time match the specified date and time criteria. CRDATE is mutually exclusive with the SINCE operand. CRDATE can be specified as any of the following:

CRDATE
Only volumes whose creation date is the current date are listed

CRDATE(START(*start_date*,*start_time*))
Only volumes whose creation date is on or after the specified start date and (if specified) start time are listed, where *start_date* is either an absolute date or relative date and *start_time* specifies the start time criteria.

CRDATE(END(*end_date,end_time*))
Only volumes whose creation date is on or before the specified end date and (if specified) end time are listed, where *end_date* is either an absolute date or relative dateand *end_time* specifies the start time criteria. Note that because START defaults to the current date, the specified end date must be equal to or greater than the current date when START is omitted.

CRDATE(START(*start_date,start_time*)END(*end_date,end_time*))
Only volumes whose creation date and creation time are within the range delimited by the specified start and end dates and times are listed, where both *start_date* and *end_date* are either an absolute date or relative date. The specified end date must be equal to or greater than the specified start date.

Each of the *start_date* and *end_date* values can be absolute or relative dates.

Absolute dates are specified as either yyyy/ddd or yyddd format. For example, January 3, 2015 can be specified as 2015/003 or 15003.

Relative Dates are specified as a number of days, months, or years prior to the current date.

-0
specifies the current day, current month, current year.

-n
specifies that the date is *n* days before the current date

-nM
specifies that the date is *n* months before the current month and the current day in the month is as the current date.

-nY
specifies that the date is *n* years before the current year and the current day in the year is as the current date.

The value range for *n* is 0 to 99999, with a required leading dash (‘-‘) and an optional suffix of M or Y.

Start and end times are specified in the *hhmmss* format, where:

hh
is hours

mm
is minutes

ss
is seconds

and assumes a 24 hour clock. For example, nine o'clock in the evening is 210000. You can specify a time in the range from 000000 to 235959.

The default start_time is 000000. The default end_time is 235959.

Note: If *start_time* is specified, then *start_date* must also be specified and if *end_time* is specified, then *end_date* must also be specified.

Examples: To list volumes whose creation date is:

Any time today
specify: SV CRDATE

Today between six o'clock and eleven o'clock in the morning
specify: SV CRDATE(START(-0,060000) END(-0,110000))

Three days ago between eleven o'clock in the morning and 5:30 in the afternoon
specify: SV CRDATE(START(-3,110000) END(-3,173000))

Before January 1, 2000
specify: SV CRDATE(START(0000/001) END(1999/365))

On or after eight o'clock in the morning, January 2, 2015
Specify: SV CRDATE(START(2015/002,080000))

CRSYSID(*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.

CRSYSID can be abbreviated as SYSID.

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 these 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 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(*system_ID*)
Specifies to list volumes where the first file was created on the named system ID.

EDM
Specify the EDM (External Data Manager) operand to select volumes with the volume EDM attribute.

The EDM field specifies whether the volume is being externally managed by the owning application. This volume attribute prevents volumes from being prematurely expired by DFSMSrmm.

EDM and NOEDM are mutually exclusive.

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. 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.

END is mutually exclusive with LIMIT.

DSNCRSYSID can be abbreviated as DSNSYSID.

EXPDT(*date_and_time_range*)
Lists the volumes whose retention expiration date and time matches the specified date criteria. 1999/365 and 1999/366 are treated as permanent retention which is greater than any other date. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHVOLUME ASDATE operand for a description of how to use the START and END suboperands and examples of their use.

FORCE(YES | NO)
Specifies a list of volumes based on the setting of the "force" attribute in the record, which is set when the CHANGEVOLUME subcommand has been used with the FORCE operand and the requested change was made only because FORCE was specified.. Specify `FORCE(YES)` to list volumes that have the "force" attribute set. Specify `FORCE(NO)` to list volumes that do **not** have the "force" attribute set.

HOLD | NOHOLD

HOLD
Specify the HOLD operand to select volumes with the volume HOLD attribute. HOLD and NOHOLD are mutually exclusive.

NOHOLD
Specify the NOHOLD operand to select volumes without the volume HOLD attribute. HOLD and NOHOLD are mutually exclusive.

HOME(SHELF |*library_name* | LOCDEF_location_name)
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.

You can enter any value, as no checking is done against the current list of locations defined to DFSMSrmm.

INSET(*value*)
Specifies a list volumes based on whether or not a volume is part of a multivolume chain or set. The INSET operand checks the previous and next volume information in the volume record. Specify one of the following values for INSET:

FIRST
List only volumes that are the first in a chain or set

LAST
List only volumes that are at the end of a multivolume set or chain.

NO
List only volumes that are **not** part of a multivolume chain or set; they have neither a previous nor a next volume.

YES
Lists all volumes that have a previous or next volume in the chain or set of volumes.

For example, to list all complete multivolume chains, use `RMM SV VOLUME(*) OWNER(*) LIMIT(*) INSET(FIRST) CLIST("RMM SV VOLUME(",") CHAIN") EXEC EXEC.RMM.CLIST`

INTRANSIT(YES | NO)
List volumes based on whether they 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 the volume destination is set. Specify INTRANSIT(YES) to list only those volumes that DFSMSrmm has identified as "in transit". Specify INTRANSIT(NO) to list only those volumes that DFSMSrmm has not identified as "in transit".

Use the INTRANSIT operand together with LOCATION to limit the list to only those volumes residing in or moving from a specific location.

JOBNAME(*jobname*)
List volumes created by the specified job name. A job name is one-to-eight alphanumeric characters or $, #, or @. You can also use a generic job name. Use % in your generic job name mask to match any one character and * to match any character string in the job name.

If you do not specify JOBNAME, the job name is not used as a selection.

If you specify JOBNAME(*), DFSMSrmm returns all volumes that match the search values specified and that have a job name. Volumes that do not have a job name are not listed. JOBNAME is mutually exclusive with NOJOBNAME.

If you have volumes with job names that include symbols other than alphanumeric and national characters, use a generic job name to find them.

LABEL(*label_type*)
Specifies to list volumes of a specific volume type. You can specify any of these values:

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

LASTCHANGEDATE(*date_and_time_range*)
Lists the volumes whose last changed date and time matches the specified date criteria. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHVOLUME ASDATE operand for a description of how to use the START and END suboperands and examples of their use.

LASTREFDATE(*date_range*)
Lists the volumes based on both the last read date and last write date, using the most recent of both dates. The most recent of the two values within the date range for a volume to be selected. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHVOLUME ASDATE operand for a description of how to use the START and END suboperands and examples of their use.

LIMIT(*search_limit*)
Specifies the number of entries that DFSMSrmm lists. The maximum allowable decimal value is 9999. More than 9999 entries can be displayed by specifying an asterisk to request a list of all entries matching your search criteria.

LIMIT is mutually exclusive with END.

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 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
Specify to list volumes residing in a specific location. Specify one of these:

Specify	To list
SHELF	Volumes stored in shelf locations in a non-system-managed library.
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.
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.
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.
generic_location_name	You can use a generic location name. You can use % in your generic mask to match any one character and * to match any character string.

MANUALMOVE
Specifies a list of volumes that have a move mode of MANUALMOVE.

MEDINF(*medinf_name*)
Specifies to list volumes with the given media information name. If you omit the MEDINF operand, no selection based on MEDINF name is performed.

You can use `MEDINF(IBM)` regardless of which MEDINF commands are defined in parmlib. Any other MEDINF name is validated against the MEDINF definitions.

Use this operand either on its own to select certain subsets of your media, or use it together with the MEDIATYPE and RECORDINGFORMAT operands to enable those operands to specify values that are either defined to DFSMSrmm by MEDINF commands in parmlib or are built in values using MEDINF(IBM).

When you do not specify MEDINF, but do specify one or more of the MEDIATYPE and RECORDINGFORMAT operands, DFSMSrmm matches the media type or recording format to the first MEDINF table entry that includes the specified external value, then to the built-in values.

Use the LISTCONTROL MEDINF subcommand to list your existing media information name entries.

MEDIANAME(*medianame*)
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 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 327 for more information.

MEDIATYPE(*media_type*)
Specifies the volume's physical media type to limit the volume search. You can specify media type values that match to those defined to DFSMSrmm by the MEDINF commands in parmlib. Use one of these values:

*****
The volume is not a cartridge. This is the default.

CST
Cartridge System Tape

ECCST
Enhanced Capacity Cartridge System Tape

EHPCT
Extended High Performance Cartridge Tape

HPCT
High Performance Cartridge Tape

MEDIA5 | ETC
IBM Enterprise Tape Cartridge

MEDIA6 | EWTC
IBM Enterprise WORM Tape Cartridge 3592

MEDIA7 | EETC
IBM Enterprise Economy Tape Cartridge 3592

MEDIA8 | EEWTC
IBM Enterprise Economy WORM Tape Cartridge 3592

MEDIA9 | EXTC
IBM Enterprise Extended Tape Cartridge 3592

MEDIA10 | EXWTC
IBM Enterprise Extended WORM Tape Cartridge 3592

MEDIA11 | EATC
IBM Enterprise Advanced Tape Cartridge

MEDIA12 | EAWTC
IBM Enterprise Advanced WORM Tape Cartridge

MEDIA13 | EAETC
IBM Enterprise Advanced Economy Tape Cartridge

MOVEDATE(*date_range*)
Lists the volumes whose movement tracking date matches the specified date criteria. The date criteria can be specified using the START and END suboperands. MOVEDATE can be specified as STOREDATE. See the description of the SEARCHVOLUME ASDATE operand for a description of how to use the START and END suboperands and examples of their use.

MOVETYPE(*type*)
Specify MOVETYPE(READYTOSCRATCH) to limit the list to volumes that are in ready to return to scratch status. Specify MOVETYPE(NOTREADYTOSCRATCH) to list volumes that are not pending release or are pending release with actions other than scratch.

NOEDM
Specify the NOEDM operand to select volumes without the volume EDM attribute.

The EDM field specifies whether the volume is being externally managed by the owning application. This volume attribute prevents volumes from being prematurely expired by DFSMSrmm.

EDM and NOEDM are mutually exclusive.

NOJOBNAME
Use this operand to list volumes that have no creating job name.

NOLIST
Specifies that DFSMSrmm should neither produce a list nor set REXX variables for resources 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.

NOOEXPDT
Specifies a list of volumes with no original expiration date. NOOEXPDT is mutually exclusive with the OEXPDT operand.

NOVOL1
Specifies a list of volumes that do not have a VOL1 recorded by DFSMSrmm.

OPEN
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.

OEXPDT(*date*)
Specifies a list of volumes based on the original expiration date. You can search using a specific date or for volumes with any original expiration date.

Specify one of the following values for OEXPDT:

***
Use an asterisk to search for volumes with any original expiration date.

date
A specific original expiration date. A specific date can be an absolute date in either *yyyy/ddd* or *yyddd* format.

Use the NOOEXPDT operand to list volumes with no original expiration date.

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 the ID of the command issuer.

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 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.

READDATE(*date_range*)
Lists the volumes whose volume last read date matches the specified date criteria. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHVOLUME ASDATE operand for a description of how to use the START and END suboperands and examples of their use.

RECORDINGFORMAT(*format_list*)
Specifies the basic recording format for tape volumes. You can specify media type values that match to those defined to DFSMSrmm by the MEDINF commands in parmlib. DFSMSrmm limits the list it returns to those volumes that match the specified value.

Specify one or more of the following values for RECORDINGFORMAT:

***
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).

EFMT2
Data has been written to the volume in Enterprise Format 2 recording technology. You can only specify EFMT2 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), MEDIATYPE(MEDIA8), MEDIATYPE(MEDIA9), and MEDIATYPE(MEDIA10).

EEFMT2
Data has been written to the volume in Enterprise Encrypted Format 2 recording technology. You can only specify EEFMT2 with MEDIATYPE(MEDIA5), MEDIATYPE(MEDIA6), MEDIATYPE(MEDIA7), MEDIATYPE(MEDIA8), MEDIATYPE(MEDIA9) and MEDIATYPE(MEDIA10).

EFMT3
Data has been written to the volume in EFMT3 (enterprise format 3) recording format. A recording format of EFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EEFMT3
Data has been written to the volume in EEFMT3 (enterprise encrypted format 3) recording format. A recording format of EEFMT3 is valid with MEDIATYPE(MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10) only.

EFMT4
Data has been written to the volume in EFMT4 (enterprise format 4) recording format. A recording format of EFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) only.

EEFMT4
Data has been written to the volume in EEFMT4 (enterprise encrypted format 4) recording format. A recording format of EEFMT4 is valid with MEDIATYPE(MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13) only.

RELEASEACTION(*action_list*)
Release actions are those that will be set as pending actions when a volume is released. Use the RELEASEACTION operand to search for volumes that have the specified release action set. Volumes are returned if any of the values you specify are set in the volume. Also see the ACTION operand for how to search for volumes with pending actions.

RELEASEACTION can also be specified as RA.

Specify one or more of the following values for RELEASEACTION, separated by commas:

ALL
To list all volumes with any release action.

ERASE
To list only volumes that require erasing.

INIT
To list only volumes that require initialization.

NOTIFY
To list only volumes for which owners notified.

REPLACE
To list only volumes that replaced by new volumes and returned to the scratch pool.

RETURN
To list only volumes that should be returned to their owner.

SCRATCH
To list all volumes to be returned to scratch status.

NOERASE
To list only volumes that do not require erasing. May be used in combination with ALL or other operands to exclude volumes that require erasing.

NOINIT
To list only volumes that do not require initialization. May be used in combination with ALL or other operands to exclude volumes that require initialization.

NONOTIFY
To list only volumes for which owners are not notified. May be used in combination with ALL or other operands to exclude volumes for which owners notified.

NOREPLACE
To list only volumes that do not need to be replaced by new volumes and returned to the scratch pool. May be used in combination with ALL or other operands to exclude volumes that replaced by new volumes and returned to the scratch pool.

NORETURN
To list only volumes that should not be returned to their owner. May be used in combination with ALL or other operands to exclude volumes that should be returned to their owner.

NOSCRATCH
To list all volumes that are not to be returned to scratch status. May be used in combination with ALL or other operands to exclude volumes to be returned to scratch status.

ONLY
Can be used in combination with other operands to limit the list to a specific combination. For example, RELEASEACTION(RETURN,NOTIFY,ONLY) will list only those volumes for which **both** the RETURN and NOTIFY release action apply. If ONLY were omitted, then DFSMSrmm would list all the volumes for which **either** release action applied.

When you specify ONLY with no other operand value, DFSMSrmm selects volumes that have no release action set. This is the same as specifying RELEASEACTION(NOERASE, NOINIT, NONOTIFY, NOREPLACE, NORETURN, NOSCRATCH).

RELEASEOPTION(*option_list*)
List volumes based on their release options. Release options are set by VRSEL processing when data sets match to, or are retained by, a VRS. Refer to Defining Release Policies for more details about how release options are handled.

RELEASEOPTION also can be specified as RO.

Specify one or more of the following values for RELEASEOPTION, separated by commas:

ALL
List all volumes that have any release options of any kind. Specifying the ALL operand value is the same as specifying RELEASEOPTION(EXPIRYDATEIGNORE,SCRATCHIMMEDIATE).

EXPIRYDATEIGNORE
List all volumes that have the "expiry date ignore" release option. EXPIRYDATEIGNORE can be specified as XI.

NOSI
List all volumes that do **not** have the "scratch immediate" release option. Can be used in combination with ALL or other operands to exclude volumes that have the "scratch immediate" release option.

NOXI
List all volumes that do **not** have the "expiry date ignore" release option. Can be used in combination with ALL or other operands to exclude volumes that have the "expiry date ignore" release option.

ONLY
Can be used in combination with other operands to limit the list to a specific combination. For example, RELEASEOPTION(SI,XI,ONLY) will list only those volumes for which **both** the SI and XI release options apply. If ONLY were omitted, then DFSMSrmm would list all the volumes for which **either** release option applied.

When you specify ONLY with no other operand value, DFSMSrmm selects volumes that have no release options set. This is the same as specifying RELEASEOPTION(NOSI,NOXI).

SCRATCHIMMEDIATE
List all volumes that have the "scratch immediate" release option. SCRATCHIMMEDIATE can be specified as SI.

REQUIRED(*required_list*)
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 these 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 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)
REQUIREDLABELVERSION(*n*) 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, 3, or 4. Specify 0 for volumes with no required label version.

There is no default.

RETAINBY(FIRSTFILE | SET | SETORFIRSTFILE | VOLUME)
RETAINBY specifies a list limited to EXPDT retained volumes based on their RETAINBY attribute. Specify:

FIRSTFILE or F
To select volumes with RETAINBY = FIRSTFILE

SET or S
To select volumes with RETAINBY = SET.

SETORFIRSTFILE
To select volumes with RETAINBY = SET or FIRSTFILE

VOLUME or V
To select volumes with RETAINBY = VOLUME.

RETAINBY can be specified as RB.

There is no default for RETAINBY. If RETAINBY is not specified, DFSMSrmm will display volumes without regard to their RETAINBY attribute.

RETDATE(*retention_date*)
RETDATE specifies that DFSMSrmm lists only MASTER and USER volumes that will expire up to and including the specified date. You can specify a specific date as RETDATE(*retention_date*). You can also specify the DFSMSrmm special date formats; CATRETPD, PERMANENT, WHILECATLG, or CYCL/*nnnnn*, where "*nnnnn*" is five numeric digits. When you specify one of the special dates, DFSMSrmm lists only those volumes that are VRS retained with that special retention date. When you specify the special cycles format date, CYCL/*nnnnn*, DFSMSrmm lists volumes that are VRS retained and have a cycles retention date and the same number or fewer cycles. For example; RETDATE(CYCL/00255) searches for all data sets with a retention date set to CYCL/00255 or lower, such as CYCL/00001. 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 2012/001. 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 12001. 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 77 for information about how DFSMSrmm calculates retention dates.

RETENTIONMETHOD(EXPDT | VRSEL)
Specifies a list limited to volumes that have a specified retention method. Specify EXPDT to select volumes with the EXPDT retention method. Specify VRSEL to select volumes with the VRSEL retention method.

There is no default for RETENTIONMETHOD. If RETENTIONMETHOD is not specified, DFSMSrmm will display volumes without regard to their retention method.

RETENTIONMETHOD can be abbreviated as RM.

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 2012/001. 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 12001. 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.

SINCE and ASDATE are mutually exclusive.

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

START
Specify this operand to request that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

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 these 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.

LOAN
To list volumes that are currently stored in loan locations.

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 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.

STORAGEGROUP(*storage_group_name*)
Specifies the storage group name in order to select a subset of volumes based on the assigned storage group name. You can enter any value, as no checking is done. 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.

DFSMSrmm accepts the abbreviation STORGRP.

Use the STORAGEGROUP operand to build lists of exportable volumes that are in the same VTS physical volume pool.

STOREDATE
An alternate keyword for MOVEDATE. See the description of the SEARCHVOLUME MOVEDATE operand.

TYPE(LOGICAL | PHYSICAL | STACKED)
Specifies a list of volumes based on volume type:

LOGICAL
Limits the list to logical volumes.

Logical volumes have a many-to-one association with physical tape media and are used indirectly by z/OS 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 through its tape volume cache or after the data has been copied to a physical volume through the use of special utilities.

PHYSICAL
Limits the list to physical volumes.

A physical volume is a volume that has a one-to-one association with physical tape media and which is used directly by z/OS applications. It can 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.

STACKED
Limits the list to stacked volumes.

A stacked volume is 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 z/OS applications, but by the virtual tape server and its associated utilities. It can be removed from a virtual tape server to allow transportation of logical volumes to a vault or to another virtual tape server.

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 if obtained logical volume resides on a stacked volume. In such a case, 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.

USE(*systems*)
Specifies a list volumes based on the systems on which the volumes can be used. Specify one or more of the following values for *systems*. If more than one value is specified, separate them with commas:

IRMM
To list volumes that can be used on open systems, managed by IRMM

ALL
To list all volumes. ALL can be used with one or more NOT*xxx* options to restrict which volumes are selected. ALL is the default.

MVS
To list volumes that can be used on MVS systems such as z/OS

NOTIRMM
To exclude volumes used on open systems managed by IRMM

NOTMVS
To exclude volumes used on MVS systems

NOTVM
To exclude volumes used on VM systems

VM
To list volumes that can be used on VM systems such as z/VM®

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.

WORM(NO|YES)
Specifies to list volumes that are identified to DFSMSrmm as WORM type volumes. Volumes are selected solely on whether they have been identified as WORM or recorded as WORM volumes. There is no selection based on media type, such as EWTC.

WRITEDATE
Lists the volumes whose volume last write date matches the specified date criteria. The date criteria can be specified using the START and END suboperands. See the description of the SEARCHVOLUME ASDATE operand for a description of how to use the START and END suboperands and examples of their use.

Task: Create a list of all duplicate volumes.

Command:

```
RMM SEARCHVOLUME VOL1(*)
```

Output: DFSMSrmm displays a list such as the one shown in Figure 110 on page 408. 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 110. 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 111 on page 408.

```
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 111. 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 112 on page 409.

```
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 112. Sample SEARCHVOLUME output listing volumes to eject

DFSMSrmm also creates a CLIST data set containing the records shown in Figure 113 on page 409.

```
RMM CHANGEVOLUME 9990A5 EJECT(BULK)
 RMM CHANGEVOLUME 9990A6 EJECT(BULK)
```

Figure 113. 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 114 on page 409.

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
 ------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
 KIM002 KIMBERLY ABC002 06/04/1993 10/01/2012 SHELF    1     M
 KIM006 KIMBERLY DAMW08 06/04/1993 10/01/2012 SHELF    1     M
 111001 LESLEY01 D65B35 06/18/1994 10/01/2012 SHELF    1     M        REMOTE
 111013 LESLEY01 D65B36 06/18/1994 10/01/2012 SHELF    1     M        REMOTE
```

Figure 114. 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 115 on page 409. DFSMSrmm displays sample CLIST output in Figure 116 on page 409.

```
A00099 LESLEY01 07/30/1999 08/26/1999 SHELF 1 M VTS1
```

Figure 115. Sample SEARCHVOLUME output building a list of logical volumes

```
CONT01,A00099
```

Figure 116. 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 117 on page 410.

```
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 117. 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 118 on page 410.

```
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 118. 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 119 on page 410.

```
RMM CHANGEVOLUME RFA199 CMOVE
```

Figure 119. 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 120 on page 411.

```
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 120. 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 121 on page 411.

```
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 121. 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 122 on page 411.

```
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 122. 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 123 on page 411 shows the output from the RMM SEARCHVOLUME subcommand request. Figure 124 on page 412 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 123. Listing volumes for the export list

```
RFA055,MAZ
RFA056,MAZ
```

Figure 124. CLIST information for the export list

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 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

Purpose

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 36. Creating lists of vital record specifications

To request the list of	You specify
Data set vital record specifications	A fully-qualified or generic data set name and optionally a job name
Volume vital record specifications	A specific or generic volume serial number
Name vital record specifications	A specific or generic vital record specification name

Use other operands to further limit your search. Specify the UNTILEXPIRED operand to restrict the search to those data sets that are 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.

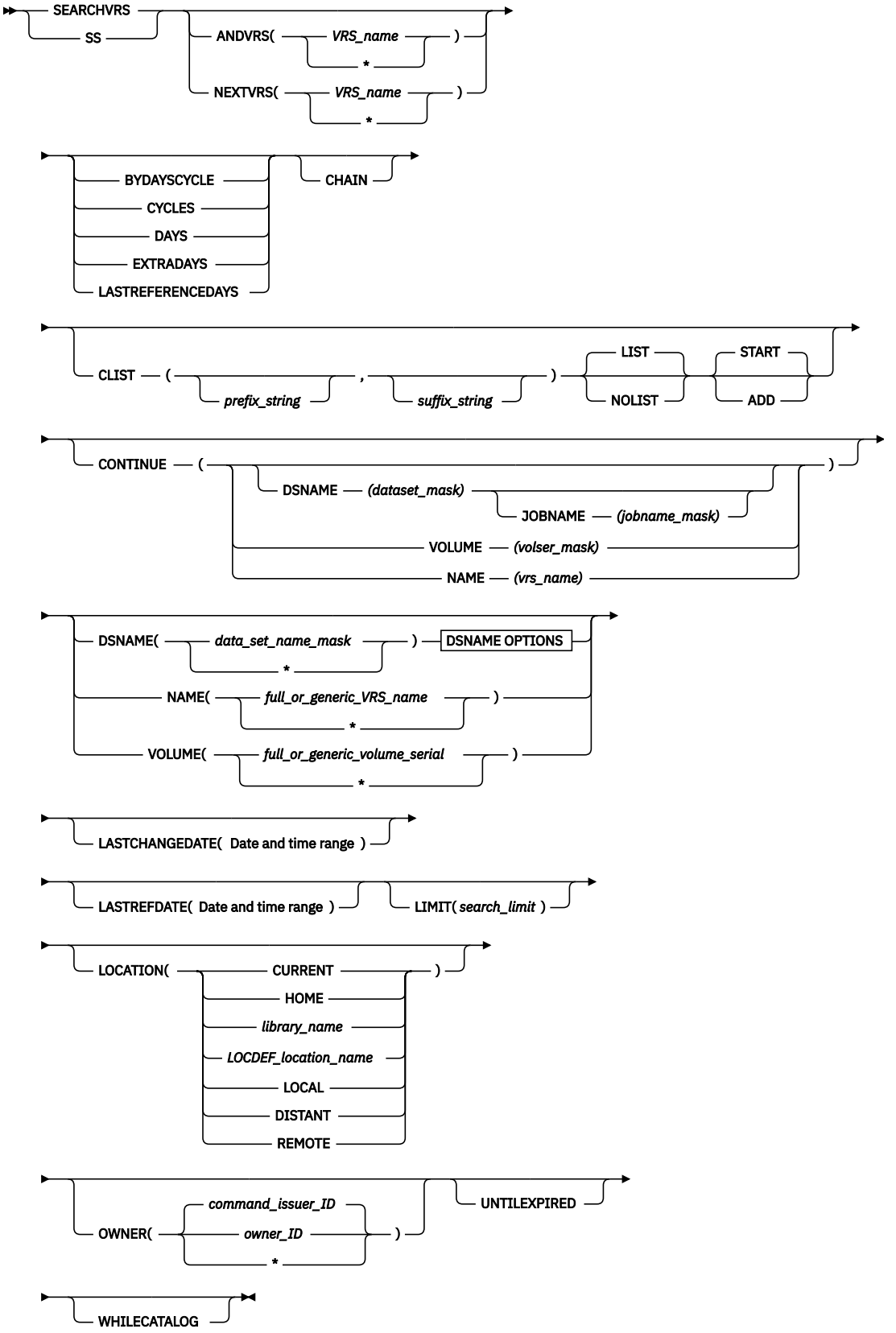
Use the LIMIT operand to restrict how many vital record specifications DFSMSrmm lists. DFSMSrmm searches for vital record specifications until it reaches the limit you specify or until it lists all vital record specifications that match your search criteria. DFSMSrmm lists a maximum of ten vital record specifications if you do not specify the LIMIT operand.

Table 37 on page 413 shows the information DFSMSrmm returns for each vital record specification in the list, in the order it is displayed:

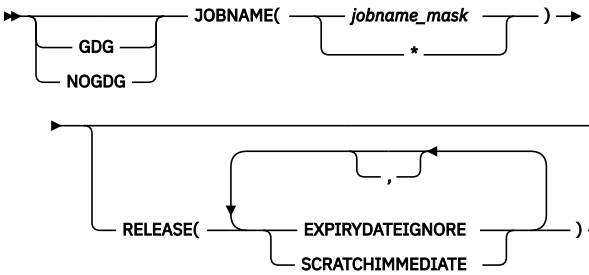
Table 37. Information returned by SEARCHVRS

Table Field Name	Description
Vital Record Specification	Volume serial number, data set name mask, or vital record specification name
Job name	Name of the job that created the data set
Type	Type of vital record specification (one of DSN, VOL or NAME)
Owner	Owner ID
Location	Location where the volume retained by the vital record specification resides
Next VRS	Name of next name vital record specification to which this vital record specification is linked

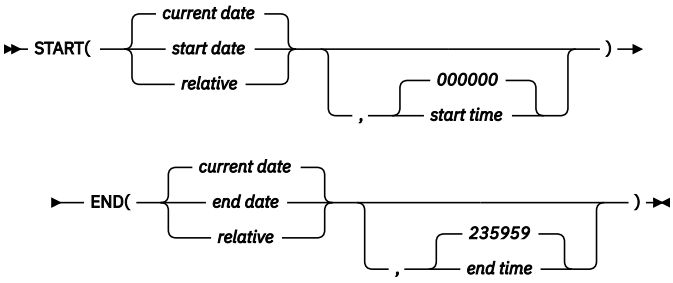
Format



DSNAME OPTIONS



Date and time range



Parameters

ADD
Specify this operand to request that new records written to the CLIST data set are added after any existing records in the data set. When the CLIST data set is empty or DFSMSrmm creates the CLIST data set during command execution, specifying ADD is the same as specifying START.

ADD is mutually exclusive with START.

You can easily build a set of commands from CLIST processing using multiple SEARCH subcommands of the same or different resource types. For variable length records, the minimum record length can cause the LRECL to be increased. For fixed length records, if the minimum length cannot be accommodated, the subcommand fails.

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 if 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 ADD, CLIST, LIST, NOLIST, START 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. To enter a null *prefix_string*, add a pair of separator characters such as '' to the text string (for example, CLIST('','*suffix_string*')).

See “Creating CLISTs of executable subcommands” on page 146 for information about the data set used for the CLIST output.

CONTINUE(*vrs_info*)
Specify the CONTINUE operand without any value to notify DFSMSrmm SEARCH subcommand processing that you want to break down the search results based on the LIMIT value and request that DFSMSrmm return the search continue information for use with the next command. For TSO, the continue information is returned either as a REXX variable or as a line mode message. When the subcommand is issued from the DFSMSrmm API, the continuation information can be either a line mode message or an SFI or XML attribute.

CONTINUE is an optional operand.

Use the LIMIT operand to control the maximum number of entries to be returned each time you start or continue the search.

To continue a previous search subcommand, the CONTINUE operand value includes one of the following *vrs_info* values to identify the current search position:

DSNAME(*dataset_mask*)
dataset_mask is one to 44 characters enclosed in single quotation marks if it contains any special characters, or blank.

JOBNAME(*jobname_mask*)
jobname_mask is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

VOLUME(*volser_mask*)
volser_mask is one to six characters enclosed in single quotation marks if it contains any special characters, or blank.

NAME(*vrs_name*)
vrs_name is one to eight characters enclosed in single quotation marks if it contains any special characters, or blank.

The information required to continue a search subcommand is returned by each search subcommand that specifies the CONTINUE operand and passed back to DFSMSrmm unchanged in order to continue the previous search. You should specify the exact same subcommand unchanged. To do this, change the CONTINUE operand value on each additional command required.

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. The name is 1 to 44 characters in length and enclosed in quotations if any special characters

are included. If the data set name is not enclosed in quotations, your TSO PROFILE PREFIX value is applied.

Note: DFSMSrmm does not fold data set names to uppercase letters when you specify quoted data set names. When you specify data set names or data set name masks, be sure to specify the correct case for each character. If you create VRS data set name masks with lowercase or mixed case letters, these do not match to data sets with all uppercase characters.

In addition to normal data set naming conventions, you can use these 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 if 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.

Period (.)
A leading or trailing period is not allowed. Consecutive periods are also not allowed.

Double asterisk (**)
Cannot be specified within a data set name qualifier.

Cannot be specified before a qualifier with a single asterisk. For example, specifying DSNAME(**.*) is not a valid data set name mask.

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 z/OS 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 might 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 @. You can specify a specific job name or a job name 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.

LASTCHANGEDATE(*date_and_time_range*)
Specifies the date and time when this VRS was last created or updated.

Use the LASTCHANGEDATE operand to specify a date and time range for your search.

A date and time range consists of the start date, start time, end date and end time, specified in the following format:

START(start_date,start_time) END(end_date,end_time)

LASTCHANGEDATE

The default for both the start date and the end date is the current date. Default start_time is 000000. Default end_time is 235959.

Each date can be an absolute date in either yyyy/ddd or yyddd format, or it can be a relative value from which DFSMSrmm calculates the date.

LASTCHANGEDATE(START(*start_date*))
You can optionally specify a starting date. If you omit the date, the current date is used. The search is performed inclusive of the start date.

LASTCHANGEDATE(END(*end_date*))
You can optionally specify an ending date. If you omit the date, the current date is used. The search is performed inclusive of the end date.

LASTCHANGEDATE(START(*start_date,start_time*)END(*end_date,end_time*))

A relative value is specified as a negative number of days, months, or years.

Absolute dates are specified in either yyyy/ddd or yyddd format. For example, January 3, 2011 can be specified as 2011/003 or 11003.

Relative Dates are specified as a number of days, months, or years before the current date.

-0
specifies the current day, current month, current year.

-n
specifies that the date is *n* days before the current date

-nM
specifies that the date is *n* months before the current month and the current day in the month is as the current date.

-nY
specifies that the date is *n* years before the current year and the current day in the year is as the current date.

The value range for *n* is 0 to 99999, with a required leading '-' and an optional suffix of M or Y.

Start and end times are specified in the *hhmmss* format, where:

hh
is hours

mm
is minutes

ss
is seconds

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

Examples: To list vital record specifications whose last changed date is:

Today
specify: SS LASTCHANGEDATE

Three days ago
specify: SS LASTCHANGEDATE(START(-3) END(-3))

Before January 1, 2000
specify: SS LASTCHANGEDATE(START(0000/001) END(1999/365))

On or after January 2, 2005
Specify: SS LASTCHANGEDATE(START(2005/002))

LASTREFDATE(*date_and_time_range*)
Lists the vital record specifications based on their last reference date and time. The last reference date of the vital record specification record is the date of the last inventory management VRSEL run that used this VRS to retain a data set or volume. See the description of the SEARCHVRS LASTCHANGEDATE operand for a description of how to use the date and time suboperands.

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(*location_name*)
Specifies to list only those vital record specifications that specify the location indicated. *location_name* can be: CURRENT, DISTANT, HOME, *library_name*, LOCAL, *LOCDEF_location_name*, or REMOTE. For example, LOCATION(CURRENT) lists only those vital record specifications that specify CURRENT as the location. *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 should neither produce a list nor set REXX variables for resources 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 the ID of the command issuer.

RELEASE(EXPIRYDATEIGNORE|SCRATCHIMMEDIATE)
Specifies that DFSMSrmm restrict the search to data set vital record specifications where EXPIRYDATEIGNORE or SCRATCHIMMEDIATE have been coded.

START
Specify this operand to request that records written to the CLIST data set start from the beginning of the data set.

START is mutually exclusive with ADD.

START is the default value.

UNTILEXPIRED
Specifies that DFSMSrmm restrict the search to vital recordspecifications where UNTILEXPIRED has been coded. The UNTILEXPIRED operand should be used with DSNAME. Using UNTILEXPIRED with the VOLUME operand returns an empty set from the search.

VOLUME(*full_or_generic volume serial*|*)
Specifies a volume serial number. A full volume serial number is one-to-six alphanumeric, 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 if it is cataloged.

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 125 on page 420.

```
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 125. 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 126 on page 421.

```
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 126. 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 127 on page 421.

```
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 127. Sample SEARCHVRS output

Return codes

See Chapter 11, “DFSMSrmm return codes and reason codes,” on page 423 for DFSMSrmm reason codes.

0
Subcommand completed normally.

4
Warning. Subcommand completed but some operands might 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 11. DFSMSrmm return codes and reason codes

This topic lists return and reason codes issued by the RMM TSO subcommand, RMM, and a set of subcommands.

RMM TSO subcommand return codes

The return codes issued by DFSMSrmm are:

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.

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 128 on page 423, 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 128. Testing RMM TSO LISTVOLUME subcommand example 1

You can also code REXX statements, as shown in Figure 129 on page 423, 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 129. Testing RMM TSO LISTVOLUME subcommand example 2

© Copyright IBM Corp. 1992, 2022

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 25 on page 187 for command abbreviations that are also used in Table 38 on page 424.

Table 38 on page 424 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-GLZ)*.

Table 38. 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 EDG3025I	SB SD SO SP SR SS SV	Search processing indicates more records might exist
04	04		SB SD SO 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	SB SR	No rack or bin numbers meet search criteria
04	08	EDG3010	SD	No data sets meet search criteria
04	08	EDG3010	SO	No owners meet search criteria
04	08	EDG3010	SP	No software products 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	EDG3921 EDG3025I	SB SD SO 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	14	EDG3026E	SB SD SO SP SR SS SV	Search subcommand interrupted by operator command.
04	16	EDG3027E	AB AR AV DB DR	Add/delete subcommand interrupted by operator command. COUNT was specified.
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.
04	278	EDG3369I	AD, CD	File expiration attributes accepted but ignored for the volume expiration because volume is retained by FIRSTFILE.

Table 38. DFSMSrmm reason codes (continued)

Return code	Reason code	Message number	Issuing command	Description
04	287	EDG3375I	AD	New data set with unknown catalog status has WHILECATALOG set by default.
04	289	EDG3377	CD	Data set already has FORCEEXPIRE set. Uncatalog not attempted.
12	00		All	I/O error on DFSMSrmm control data set
12	ALL	EDG3013	AV	DFSMSrmm issues values for variables EDG@CNT, EDG@RCK, and EDG@VOL when COUNT is greater than 1 and the command fails.
12	02	EDG3220	CO	Incorrect node and user ID combination
12	04	EDG3005	All	Issued when subcommand issued incorrectly
12	08	EDG3207	All	Subsystem does not exist
12	10	EDG3208	All	Disastrous error during subsystem processing
12	12	EDG3209	All	Logical error during subsystem processing
12	14	EDG3221	AD	Incorrect request for a scratch volume
12	14	EDG3221	CV	Incorrect request for a scratch volume
12	16	EDG3222	DO	Owner owns volumes but no new owner specified
12	18	EDG3017 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.
12	38	EDG3229	AV CV DB DR	Rack number does not exist or is not empty

Table 38. DFSMSrmm reason codes (continued)

Return code	Reason code	Message number	Issuing command	Description
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
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

Table 38. DFSMSrmm reason codes (continued)

Return code	Reason code	Message number	Issuing command	Description
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	AD, CD	Installation defined maximum retention period exceeded.
12	110	EDG3023	CV DV	Volume pending release
12	114	EDG3265	AS	First STORENUMBER cannot be 99999
12	116	EDG3268	SB SD SO SP SR SS SV	Unable to open CLIST data set
12	118	EDG3269	SB SD SO SP SR SS SV	CLIST organization not sequential or partitioned
12	120	EDG3270	SB SD SO 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.
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.

Table 38. DFSMSrmm reason codes (continued)

Return code	Reason code	Message number	Issuing command	Description
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 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.
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

Table 38. DFSMSrmm reason codes (continued)

Return code	Reason code	Message number	Issuing command	Description
12	242	EDG3356	DV	Delete volume rejected; TCDB status does not match DFSMSrmm volume status
12	244	EDG3357I	CV	WWID cannot be changed once it is set.
12	246	EDG3359E	AV, CV	Inconsistent media information.
12	248	EDG3358E	DV	Volume not pending replace.
12	250	EDG3215I	GV	GV request rejected. Another GV request is in progress, and your request is cancelled.
12	254	EDG3360E	SB SD SO SP SR SS SV	A CLIST data set with fixed LRECL has a too short record length.
12	256	EDG3216	All	DFSMSrmm is quiesced.
12	258	EDG3361	CV DV	Volume HOLD attribute is set.
12	260	EDG3362	CV	Volume HOLD attribute is not set.
12	262	EDG2002I	All	DFSMSrmm subsystem cancelled by operator
12	264	EDG2003E	All	Subcommand ended abnormally
12	266	EDG3363	CV AV	Retention method can be specified only for the first volume in a set.
12	268	EDG3364	CD	VRSELEXCLUDE(NO) can be specified only for a data set in a volume set managed by VRSEL retention method.
12	270	EDG3365	CD	COPYFROM data set does not exist.
12	272	EDG3366	CD	COPYFROM validation failed for LRECL/RECFM between source and target data sets.
12	274	EDG3367	AD, CD	LASTREF and NOLASTREF can be specified only for data sets on a volume retained by the EXPDT retention method.
12	276	EDG3368	AD, CD, CV	FORCE operand required.
12	280	EDG3370	AV, CV	RETAINBY only valid with RM(EXPDT).
12	282	EDG3371	AV, CV	RETAINBY can be specified only for a the first volume in a set.
12	284	EDG3372	CV	Change of expiration date not allowed for volumes retained by first file.
12	285	EDG3373E	AD	WHILECATALOG(ON) and WHILECATALOG(UNTILEXPIRED) can be specified only for data sets on a volume retained by the EXPDT retention method.
12	286	EDG3374E	CD	WHILECATALOG(UNTILEXPIRED) specified, but catalog status is not YES
12	288	EDG3376	DV	The DELETEVOLUME subcommand specified with the RELEASE operand failed because the volume to be deleted is externally managed.
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.

Table 38. DFSMSrmm reason codes (continued)

Return code	Reason code	Message number	Issuing command	Description
20	14		AS SS	DSNAME value does not meet DFSMSrmm generic naming restrictions.
20	16	EDG3272	CV	HOME operand specified a location 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.
20	24		AS DS LS SD SS SV	JOBNAME value does not adhere to DFSMSrmm generic naming rules.
20	26		SD	PROGRAMNAME/LASTPROGRAMNAME value does not adhere to DFSMSrmm generic 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.
20	42		AV, CV	Specified media information not defined.
20	44		SV	LOCATION value does not adhere to DFSMSrmm generic naming rules.
20	46		AD AS AV CD CV	Data set name failed data set name rules check.
28	0	EDG3028	All	Attention key (ATTN) interrupted DFSMSrmm subsystem request processing. The RMM TSO command processor cannot determine whether command completed or not.

Chapter 12. DFSMSrmm REXX variables

This topic 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.

To get REXX variables for the date and time returned in a selected time zone, other than the system local time zone, you must set the REXX variable SYSAUTH.EDGTZ to a valid time zone offset. Default values are returned in local time.

The REXX variable SYSAUTH.EDGTZ has the same format as the value for the TZ subcommand operand:

```
TZ({+|-}HH[:MM[:SS]])
```

Specifies the time zone offset when date and time values are specified. The format is *{+|-}HH[:MM[:SS]]* where:

- *+/-* is the offset direction. Specify + to indicate that the offset is East of the zero median (UT). Specify - to indicate that the offset is West of the zero median (UT). The offset direction is required.
- *HH* is hours
- *MM* is minutes
- *SS* is seconds
- An optional colon (:) separates hours from optional minutes and optional seconds.
- You can specify a time in the range between 00:00:00 to 15:00:00 for HH:MM:SS. MM and SS value range is 00 to 59.

See *z/OS DFSMSrmm Implementation and Customization Guide* for information on SYSAUTH.EDGTZ and setting up DFSMSrmm common time support.

All commands set EDG@RC if REXX special variable, RC, is 4, 12, or 20.

When you specify the NOLIST and CLIST operands on a SEARCH subcommand, DFSMSrmm skips the creation of REXX variables for matching resources, except for the EDG@RC special REXX variable.

Some stem variables use the stem value of 0 to indicate the number of items returned by the command for that variable.

Multiple resources stem variables

In some cases, such as EDG@VOL (for SEARCHVOLUME), the .0 stem variable indicates that multiple resources meet the search criteria. 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. These variables are listed as "Multiple resources" stem variables in Table 39 on page 432.

Another example of a multiple resource stem variable is EDG@DSN (for SEARCHDATASET). If you issue the RMM SEARCHDATASET subcommand, EDG@DSN.0 will contain the number of returned resources, which is the number of data sets that met the search criteria. EDG@DSN.1 will contain the data set name for the first data set found and EDG@VOL.1 will contain the volser for the first data set found.

Note: For SEARCHDATASET, only EDG@DSN.0 will contain the number of found data sets.

If you want to see what data is provided in the variables that a subcommand returns, you can use a REXX exec like this to list each variable and the returned output (you will need to adjust the sysauth.edgdate value to suit your environment):

```
/* REXX */
   sysauth.edgdate = 'AMERICAN'
```

© Copyright IBM Corp. 1992, 2022

```
   "RMM SD OWNER(*)"
   say 'edg@dsn.0 =' edg@dsn.0
   do i=1 to edg@dsn.0
     say 'edg@dsn.'i ' =' edg@dsn.i
     say 'edg@vol.'i ' =' edg@vol.i
     say 'edg@own.'i ' =' edg@own.i
     say 'edg@cdt.'i ' =' edg@cdt.i
     say 'edg@file.'i ' =' edg@file.i
   end
```

The output will look like this:

```
edg@dsn.0 = 2
   edg@dsn.1 = BERNDS.DATA.SET1
   edg@vol.1 = ELW120
   edg@own.1 = D008210
   edg@cdt.1 = 06/26/2013
   edg@file.1 = 1
   edg@dsn.2 = BERNDS.DATA.SET2
   edg@vol.2 = ELW120
   edg@own.2 = D008210
   edg@cdt.2 = 06/27/2013
   edg@file.2 = 2
```

Multiple values stem variables

In other cases, such as EDG@VOL (for LISTPRODUCT), the .0 stem variable indicates how many of some repeatable value exist for a single resource. For example, if you issue the RMM LISTPRODUCT subcommand, EDGVOL.0 might contain 5, indicating that five volume serial numbers are associated with the listed product. EDG@VOL.1 contains the first volume serial number, and EDG@VOL.2 contains the second volume serial number, and so on. These variables are listed as "Multiple values" stem variables in Table 39 on page 432.

Double stem variables

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. These variables are listed as "double stem" variables in Table 39 on page 432.

TSO subcommand variables by subcommand

Table 39 on page 432 lists all the variables in RMM subcommand order that you can use in your REXX execs.

Table 39. TSO subcommand variables by RMM subcommand

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
ADDBIN	EDG@CNT EDG@RC EDG@RCK				
ADDDATASET	EDG@RC				
ADDOWNER	EDG@RC				
ADDPRODUCT	EDG@RC				
ADDRACK	EDG@CNT EDG@RC EDG@RCK				

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
ADDVOLUME	EDG@CLIB EDG@CNT EDG@CSG EDG@FRC EDG@FRS EDG@RC EDG@RCK EDG@VOL				
ADDVRS	EDG@RC				
CHANGEDATASET	EDG@RC				
CHANGEOWNER	EDG@RC				
CHANGEPRODUCT	EDG@RC				
CHANGEVOLUME	EDG@CLIB EDG@CSG EDG@FRC EDG@FRS EDG@MEDN EDG@RC EDG@RCK EDG@VOL				
CHANGEVRS	EDG@RC				
DELETEBIN	EDG@CNT EDG@RC EDG@RCK				
DELETEDATASET	EDG@RC				
DELETEOWNER	EDG@RC				
DELETEPRODUCT	EDG@RC				
DELETERACK	EDG@CNT EDG@RC EDG@RCK				
DELETEVOLUME	EDG@FRC EDG@FRS EDG@RC				
DELETEVRS	EDG@RC				
GETVOLUME	EDG@CLIB EDG@FRC EDG@FRS EDG@OWN EDG@RC EDG@VOL				
LISTBIN	EDG@LCDT EDG@LCID EDG@LCSI EDG@LCTM EDG@LCUD EDG@LCUT EDG@LOC EDG@MEDN EDG@MIV EDG@MOV EDG@OVOL EDG@PID EDG@RC EDG@RCK EDG@RST EDG@VOL				
LISTCONTROL ACTIONS	EDG@RC	EDG@ACT EDG@AST			

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
LISTCONTROL CNTL	EDG@BDT EDG@BTM EDG@CDSF EDG@CDSQ EDG@CDSU EDG@CSDT EDG@CSHN EDG@CSIP EDG@CSTM EDG@CSVE EDG@DBN EDG@DDT 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@FVP EDG@FXP EDG@JBDT EDG@JBTM EDG@JRNF EDG@JRNS EDG@JRNU EDG@LBN EDG@LRK EDG@MDT EDG@MTP EDG@MTM EDG@RBN EDG@RC EDG@RDT EDG@RMID EDG@RTM EDG@SOSD EDG@SOST EDG@UDT EDG@UTC EDG@UTM EDG@VDT EDG@VTM EDG@X100 EDG@X200 EDG@X300 EDG@XDT EDG@XTM				
LISTCONTROL DEFTABLE	EDG@RC	EDG@DEFD EDG@DEFE EDG@DEFJ EDG@DEFK EDG@DEFL EDG@DEFM EDG@DEFN EDG@DEFO EDG@DEFP EDG@DEFR EDG@DEFV EDG@DEFW EDG@DEFX EDG@DEFY EDG@DEFZ			
LISTCONTROL LOCDEF	EDG@RC	EDG@LDAM EDG@LDDF EDG@LDLT EDG@LDMT EDG@LDPR		EDG@LDLC	EDG@LDMN
LISTCONTROL MNTMSG	EDG@RC	EDG@OPL EDG@OVL EDG@SMI		EDG@MID	
LISTCONTROL MEDINF	EDG@RC	EDG@MDRA EDG@MDRP EDG@MDRT EDG@MDRW EDG@MDRX EDG@MDTX EDG@MEDR EDG@MEDT EDG@VCAP		EDG@MDNF	

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
LISTCONTROL MOVES	EDG@RC	EDG@MFR EDG@MST EDG@MTO EDG@MTY		EDG@MFR	
LISTCONTROL OPENRULE	EDG@RC	EDG@ORCI EDG@ORCO EDG@ORIA EDG@ORII EDG@ORIR EDG@ORIS EDG@OROA EDG@OROI EDG@OROR EDG@OROS EDG@ORVE EDG@ORVL EDG@ORVS		EDG@ORTP	

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
LISTCONTROL OPTION	EDG@ACCT EDG@ACS EDG@AUD EDG@BKPP EDG@BLP EDG@CATS EDG@CAT1 EDG@CAT2 EDG@CAT3 EDG@CAT4 EDG@CDS EDG@CMDD EDG@CMDO EDG@CRP EDG@CTLD EDG@DEFC EDG@DRP EDG@DSFX EDG@DSPD EDG@DSPM EDG@DSXP EDG@DTE EDG@EDM EDG@EXRB EDG@GDGC EDG@GDRP EDG@GWCT EDG@GDGD EDG@HSKP EDG@IPL EDG@JDS EDG@JRNF EDG@JRNT EDG@LCT EDG@LCTK EDG@LRED EDG@MCAT EDG@MDS EDG@MEDN EDG@MOP EDG@MRP EDG@MSGF EDG@MVBY EDG@NDRP EDG@NWCT EDG@NOT EDG@OPM EDG@PACS EDG@PDA EDG@PDAC EDG@PDAL EDG@PDAS EDG@PSF2 EDG@PSFX EDG@RC EDG@RCF EDG@RM EDG@RTBY EDG@RUB EDG@SID EDG@SLM EDG@SMP EDG@SMUC EDG@SMUE EDG@SMUS EDG@SOSP EDG@SRHN EDG@SRIP EDG@SRPN EDG@SRTK EDG@SSM EDG@SSTY EDG@TVXD EDG@TVXP EDG@UNC EDG@VACT EDG@VCHG EDG@VDRA EDG@VDRC EDG@VDRP EDG@VMIN EDG@VREA EDG@VREC EDG@VREP EDG@VRJ EDG@VRSL EDG@XDRA EDG@XDRC EDG@XDRP				
LISTCONTROL PRTITION	EDG@RC	EDG@PTNA EDG@PTNL EDG@PTSA EDG@PTVE EDG@PTVL EDG@PTVS		EDG@PTTP	
LISTCONTROL REJECT	EDG@RC	EDG@TAC		EDG@GRK	
LISTCONTROL SECCLS	EDG@RC	EDG@ERS EDG@MSG EDG@NME EDG@SEC EDG@SMF		EDG@CLS	

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
LISTCONTROL SECLEVEL	EDG@CLS EDG@ERS EDG@MSG EDG@NME EDG@RC EDG@SEC EDG@SMF			EDG@DNM	
LISTCONTROL STATUS	EDG@JRNS EDG@RC EDG@RMID EDG@STDS EDG@STLA EDG@STLH EDG@STLO EDG@STLR EDG@STNH EDG@STPL EDG@STQC EDG@STQN EDG@STQR EDG@STRH EDG@STRM EDG@STSA EDG@STSH EDG@STSL EDG@STSO	EDG@STIS EDG@STIT EDG@STIV EDG@STRF EDG@STRT EDG@STST EDG@STTQ EDG@STTR EDG@STTS		EDG@STTT	
LISTCONTROL VLPOOL	EDG@RC	EDG@ACT EDG@MEDN EDG@MOP EDG@PDS EDG@PLN EDG@PRF EDG@PSN EDG@PTP EDG@SCRM EDG@XDC		EDG@PID	
LISTDATASET	EDG@ABND EDG@BESK EDG@BLKC EDG@BLKS EDG@BLKT EDG@BLK6 EDG@CDT EDG@CJBN EDG@CLS EDG@CPGM EDG@CRAT EDG@CTLG EDG@CTM EDG@CTRT EDG@DC EDG@DD EDG@DEV EDG@DLR EDG@DLTD EDG@DLW EDG@DPCT EDG@DSEQ EDG@DSN EDG@DSS6 EDG@FILE EDG@FRXP EDG@LCDT EDG@LCID EDG@LCSI EDG@LCTM EDG@LCUD EDG@LCUT EDG@LDD EDG@LDEV EDG@LJOB EDG@LPGM EDG@LRCL EDG@LREDEDG@LSTP EDG@MC EDG@NME EDG@OWN EDG@OXD EDG@PSZ6 EDG@RC EDG@RCFM EDG@RTDT EDG@SC EDG@SG EDG@STEP EDG@SYS EDG@VEX EDG@VJBN EDG@VMV EDG@VNME EDG@VOL EDG@VRSR EDG@VSCD EDG@VSCN EDG@VTYP EDG@WCTL EDG@XDSB EDG@XDT EDG@XTM EDG@2JBN EDG@2NME EDG@2SCD EDG@2SCN				

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
LISTOWNER	EDG@AD1 EDG@AD2 EDG@AD3 EDG@DPT EDG@EML EDG@EMN EDG@EMU EDG@ETL EDG@FOR EDG@ITL EDG@LCDT EDG@LCID EDG@LCSI EDG@LCTM EDG@LCUD EDG@LCUT EDG@OWN EDG@RC EDG@SUR EDG@VLN				
LISTPRODUCT	EDG@LCDT EDG@LCID EDG@LCSI EDG@LCTM EDG@LCUD EDG@LCUT EDG@OWN EDG@PDSC EDG@PNME EDG@PNUM EDG@RC EDG@VER	EDG@FCD EDG@RCK		EDG@VOL	
LISTRACK	EDG@LCDT EDG@LCID EDG@LCSI EDG@LCTM EDG@LCUD EDG@LCUT EDG@LOC EDG@MEDN EDG@MIV EDG@MOV EDG@OVOL EDG@PID EDG@RC EDG@RCK EDG@RST EDG@VOL				
LISTVOLUME ACCESS	EDG@ID1-12 EDG@IRMM EDG@LCDT EDG@LCID EDG@LCSI EDG@LCTM EDG@LCUD EDG@LCUT EDG@MVS EDG@OAC EDG@RC EDG@VAC EDG@VM				

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
LISTVOLUME VOL	EDG@ACN EDG@ACT EDG@ADT EDG@ATM EDG@AVL EDG@CDT EDG@CLS EDG@CRID EDG@CTM EDG@CTRT EDG@DEN EDG@DESC EDG@DKBC EDG@DSEQ EDG@DSN EDG@DSYS EDG@EDM EDG@EXRB EDG@HLD EDG@JOB EDG@KEL1 EDG@KEL2 EDG@KEM1 EDG@KEM2 EDG@LBL EDG@LOAN EDG@LVC EDG@LVN EDG@MDNF EDG@MEDA EDG@MEDC EDG@MEDR EDG@MEDT EDG@NME EDG@OCE EDG@OLON EDG@OWN EDG@OXD EDG@PEND EDG@RBYS EDG@RC EDG@RCK EDG@RM EDG@RMSB EDG@RTDT EDG@SGN EDG@STVC EDG@SYS EDG@VNDR EDG@VOL EDG@VOL1 EDG@VOLT EDG@VRSI EDG@VRXI EDG@VST EDG@WORM EDG@WWID EDG@XDSB EDG@XDT EDG@XTM				
LISTVOLUME STATS	EDG@CRAT EDG@DLR EDG@DLW EDG@DSC EDG@DSR EDG@FCD EDG@LDEV EDG@MEDN EDG@NVL EDG@PNUM EDG@PRD EDG@PSZ6 EDG@PVL EDG@PWT EDG@RC EDG@SEQ EDG@TRD EDG@TWT EDG@USE6 EDG@USEC EDG@USEM EDG@VCAP EDG@VER EDG@VPCT EDG@VWMC				
LISTVOLUME STORE	EDG@BIN EDG@BMN EDG@CTNR EDG@DBIN EDG@DBMN EDG@DEST EDG@DSTT EDG@HLOC EDG@HLOT EDG@INTR EDG@LOC EDG@LOCT EDG@MOVM EDG@NLOC EDG@NLOT EDG@OBMN EDG@OBN EDG@OLOC EDG@OLOT EDG@RC EDG@RLPR EDG@SDT				

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
LISTVRS	EDG@DDT EDG@DESC EDG@DLR EDG@DSN EDG@LCDT EDG@LCID EDG@LCSI EDG@LCTM EDG@LCUD EDG@LCUT EDG@LOC EDG@NAME EDG@NVRS EDG@OWN EDG@PRTY EDG@RC EDG@RET EDG@RWC EDG@SC1 EDG@TLR EDG@TYP EDG@UEX EDG@VANX EDG@VDD EDG@VJBN EDG@VOL EDG@VRC EDG@VRSI EDG@VRXI				
SEARCHBIN	EDG@CONT EDG@RC	EDG@LOC EDG@MEDN EDG@MIV EDG@MOV EDG@OVOL EDG@PID EDG@RCK EDG@RST EDG@VOL	EDG@RCK		
SEARCHDATASET	EDG@CONT EDG@RC	EDG@CDT EDG@CDTJ EDG@CJBN EDG@CTM EDG@FILE EDG@KEYF EDG@KEYT EDG@LRED EDG@OWN EDG@OXD EDG@RTDJ EDG@RTDT EDG@TYPF EDG@TYPT EDG@VOL EDG@VRSR EDG@XDT EDG@XDTJ	EDG@DSN		
SEARCHOWNER	EDG@CONT EDG@RC	EDG@AD1 EDG@AD2 EDG@AD3 EDG@DPT EDG@EML EDG@EMN EDG@EMU EDG@ETL EDG@FOR EDG@ITL EDG@SUR EDG@VLN	EDG@OWN		
SEARCHPRODUCT	EDG@CONT EDG@RC	EDG@FCD EDG@OWN EDG@PDSC EDG@PNME EDG@VER EDG@VLN EDG@VOL	EDG@PNUM		

Table 39. TSO subcommand variables by RMM subcommand (continued)

Subcommand	Variables	Stem Variables			
		Single resource / value	Multiple resources (see "Multiple resources stem variables" on page 431)	Multiple values (see "Multiple values stem variables" on page 432)	Double-Stem (see "Double stem variables" on page 432)
SEARCHRACK	EDG@CONT EDG@RC	EDG@LOC EDG@MEDN EDG@MIV EDG@MOV EDG@OVOL EDG@PID EDG@RST EDG@VOL	EDG@RCK		
SEARCHVOLUME	EDG@CONT EDG@RC	EDG@ADT EDG@ADTJ EDG@ATM EDG@AVL EDG@DEST EDG@DSC EDG@DSYS EDG@EXRB EDG@HLOC EDG@INTR EDG@KEYF EDG@KEYT EDG@LBL EDG@LOAN EDG@LOC EDG@LVC EDG@LVN EDG@MEDA EDG@MEDC EDG@MEDN EDG@MEDR EDG@MEDT EDG@OWN EDG@PEND EDG@RBYS EDG@RCK EDG@RTDJ EDG@RTDT EDG@SYS EDG@TYPF EDG@TYPT EDG@VST EDG@XDT EDG@XDTJ	EDG@VOL		
SEARCHVRS	EDG@CONT EDG@RC	EDG@DDT EDG@DDTJ EDG@DLR EDG@DLRJ EDG@LOC EDG@NVRS EDG@OWN EDG@PRTY EDG@RET EDG@RWC EDG@SC1 EDG@TLR EDG@TYP EDG@UEX EDG@VANX EDG@VJBN EDG@VRC EDG@VRS EDG@VRSI EDG@VRXI	EDG@VRS		

Notes:

1. Values are always returned and contain the built-in value, if REPLACE operand was not specified.

TSO subcommand variables by name

Table 40 on page 442 lists the variables you can use in your REXX execs by variable name. See Table 25 on page 187 for command abbreviations that are used in this topic.

Table 40. TSO subcommand variables by name

Variable Name	Abbrev 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 E=ERASE I=INIT N=NOTIFY O=OWNER R=REPLACE S=SCRATCH	One of O, OE, OEN, OI, OIE, OIEN, OIN, ON, R, RE, REN, RI, RIE, RIEN, RIN, RN, S, SE, SEN, SI, SIE, SIEN, SIN, SN
	LC"1" on page 466	Action name	One of ERASE, INIT, NOTIFY, REPLACE, or RETURN
	LC VLPOOL"1" on page 466	Action name	N or blank
EDG@ADT	LV SV"1" on page 466	Assigned date	Date format
EDG@ADTJ	SV"1" on page 466	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" on page 466	Action status	One of Pending, Confirmed, Complete, or Unknown
EDG@ATM	LV SV"1" on page 466	Assigned time	8 characters (hh:mm:ss)
EDG@AUD	LC	SMF audit record type	Numeric: 0, 42, 128 - 255. 0 is no audit records written; 42, 128 - 255 are the audit record types
EDG@AVL	LV SV"1" on page 466	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@BESK	LD SD"1" on page 466	BES key index	Numeric: 0 - 4294967295

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	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: 10 characters
EDG@BLKS	LD SD"1" on page 466	Size of data set blocks	Numeric: 1 - 999999
EDG@BLKT	LD	Total block count	Up to 10 numeric characters. A value of "-1" indicates that the BLK6 variable should be used.
EDG@BLK6	LD	Total block count	Numeric: 1 - 18446744073709551615
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@CAT1	LC OPTION	CATSYSID list(1-4)	35 characters
EDG@CAT2	LC OPTION	CATSYSID list(5-8)	35 characters
EDG@CAT3	LC OPTION	CATSYSID list(9-12)	35 characters
EDG@CAT4	LC OPTION	CATSYSID list(13-16)	35 characters
EDG@CDS	LC	Control data set identifier	8 characters
EDG@CDSF	LC CNTL	CDSFULL parmlib operand value	Numeric: 0-99
EDG@CDT	LD LV SD"1" on page 466	Volume or data set create date	Date format
EDG@CDSQ	LC CNTL	Control data set ENQ	Disabled or Enabled
EDG@CDSU	LC CNTL	Control data set percentage used	Numeric: 0 - 100
EDG@CDTJ	SD"1" on page 466	Data set create date	Julian date format
EDG@CJBN	LD	Job name	8 characters
	SD"1" on page 466	Job name	8 characters
EDG@CLIB	AV"14" on page 466 CV"14" on page 466 DV"14" on page 466	Current library name	8 characters
EDG@CLS	LC"2" on page 466 LD LV	Security classification description	32 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@CMDD	LC OPT	Command authorization based on data set name	3 characters: YES or NO
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@CONT	All SEARCH subcommands	SEARCH Continue information	84 characters
EDG@CPGM	LD	Creating program name	8 characters
EDG@CRAT	LD LV	Compression ratio	Numeric: 0.00 – 999.99 There will always be 2 decimal places.
EDG@CRID	LV	Create user ID	8 characters
EDG@CRP	LC	CATRETPD retention period	Numeric: 0 - 9999
EDG@CSDT	LC CNTL	Catalog synchronize date	Date format
EDG@CSG	AV"13" on page 466 CV"13" on page 466	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	45 characters: 1 to 45 hexadecimal characters including colon, 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@CTLD	LC OPT	Catalog days	Numeric: 0 - 93000
EDG@CTLG	LD	Catalog status	One of UNKNOWN, YES, or NO
EDG@CTM	LD SD"1" on page 466	Data set create time	8 characters (hh:mm:ss)
	LV	Volume create time	8 characters (hh:mm:ss)
EDG@CTNR	LV	Container in which the resource is stored (stacked volume)	6 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@CTRT	LD LV	Catalog Retained	blank, 'KeptByCatalog', 'UntilExpired', 'ForceExpire'
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” on page 466	Vital record specification deletion date	Date format
EDG@DDTJ	SS“1” on page 466	Vital record specification delete date	Julian date format
EDG@DEFC	LC OPT	Defaults table entries count	numeric: 0-32768
EDG@DEFD	LC DEFTABLE[1]	Defaults table data set name mask	44 characters
EDG@DEFE	LC DEFTABLE[1]	Defaults table retention period default	numeric: 0-93000 or PERM or blank
EDG@DEFJ	LC DEFTABLE[1]	Defaults table jobname mask	8 characters
EDG@DEFK	LC DEFTABLE[1]	Defaults table keydate mask	5 characters of the form yyddd or NOKEYDATE or NODATE or *
EDG@DEFL	LC DEFTABLE[1]	Defaults table last reference extra days default	numeric: 0-93000 or blank
EDG@DEFM	LC DEFTABLE[1]	Defaults table EDM. Indicates volume expiration is managed by an External Data Manager	Y or N or blank
EDG@DEFN	LC DEFTABLE[1]	Defaults table program name mask	8 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@DEFO	LC DEFTABLE[1]	Defaults table retention override - indicates whether RETPD overrides the value specified in the JCL	Y or N or blank
EDG@DEFP	LC DEFTABLE[1]	Defaults table scratch pool	8 characters
EDG@DEFR	LC DEFTABLE[1]	Defaults table retention method default	VRSEL, EXPDT, or blank
EDG@DEFV	LC DEFTABLE[1]	Defaults table VRS management value	8 characters
EDG@DEFW	LC DEFTABLE[1]	Defaults table WHILECATALOG default	OFF, ON, UX, or blank
EDG@DEFX	LC DEFTABLE[1]	Defaults table VRSEL exclude default	Y, N, or blank
EDG@DEFY	LC DEFTABLE[1]	Defaults table RETAINBY default	9 characters: VOLUME, SET, FIRSTFILE or blank
EDG@DEFZ	LC DEFTABLE[1]	Defaults table continue, indicates whether we need to continue searching the defaults table to see if any other entries match	Y or blank
EDG@DEN	LV	Volume density	One of 1600, 6250, 3480, or IDRC, or * (undefined)
EDG@DESC	LV	Volume description	30 characters
	LS	Vital record specification description	30 characters
EDG@DEST	LV SV"1" on page 466	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@DKBC	LV	Data sets kept by catalog	Numeric: 1 - 99999
EDG@DLR	LD LS SS"1" on page 466	Date data set on volume last referenced/read	Date format
EDG@DLRJ	SS"1" on page 466	Last reference date	Julian date format
EDG@DLTD	LD	Deleted by disposition processing	YES or NO
EDG@DLW	LV LD	Date data set on volume last written to	Date format

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@DNM	LC"1" on page 466	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 - 93000, PERMANENT
EDG@DSC	LV SV"1" on page 466	Number of data sets on a volume	Numeric: 0 - 65535
EDG@DSEQ	LD LV	Data set sequence number	Numeric: 0 - 65535
EDG@DSFX	LC OPT	Active defaults table	2 characters
EDG@DSN	LD LV LS"8" on page 466 SD"1" on page 466	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" on page 466	Data set recording	ON or OFF
EDG@DSS6	LD	Data set size	Factored number FF*num*. Up to 12 characters: FF is KB, MB, GB, TB *num* is numeric 0 - 9999999999
EDG@DSTT	LV	Destination type	One of AUTO, MANUAL, STORE, or blank
EDG@DSXP	LC OPT	DSNEXPIRE option	NONE, BLOCK
EDG@DSYS	LV VOL, SV"1" on page 466	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@EBIN	LC CNTL	Extended bin enable status	8 characters: one of ENABLED, DISABLED
EDG@EDM	LC OPT, LV	EDM attribute	1 character: N or Y
EDG@EML	LO, SO"1" on page 466	Owner's e-mail address	1 to 63 characters
EDG@EMN	LO	Owner's node	8 characters
EDG@EMU	LO	Owner's user ID	8 characters
EDG@ERS	LC"2" on page 466	Security classification erase option	Y or N
EDG@ETL	LO	Owner's external telephone number	20 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@EXRB	LC OPT, LV VOL, SV"1" on page 466	RETENTIONMETHOD EXPDT RETAINBY	9 characters: one of FIRSTFILE, SET, or VOLUME
EDG@FBP	LC	Control data set 'backup in progress' flag	Y or N
EDG@FCD	LP"1" on page 466 LV SP"1" on page 466	Software product feature code	4 characters
EDG@FCSP	LC CNTL	Catalog Synchronize in progress	Y or N
EDG@FDB	LC	Number of free bin numbers in DISTANT storage location	Numeric: 0 - 999999
EDG@FEP	LC	Report extract processing in progress flag	Y or N
EDG@FILE	LD SD"1" on page 466	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@FRXP	LD	FORCEEXPIRE flag	NO, YES
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

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@FXP	LC	Expiration processing in progress flag	Y or N
EDG@GDGC	LC OPT	GDG cycleby	10 characters: CRDATE or GENERATION
EDG@GDGD	LC OPT	GDG duplicate	5 characters: BUMP, COUNT, DROP, or KEEP
EDG@GDRP	LC OPT	Default retention period for RM=EXPDT GDG data sets	Numeric: 0 -93000 or 11 characters: PERMANENT
EDG@GRK	LC"1" on page 466	Generic rack number	6 characters
EDG@GWCT	LC OPT	Default WHILECATALOG setting for RM=EXPDT GDG data sets	12 characters: OFF, ON, or UntilExpired
EDG@HLD	LV	Hold attribute	1 character: N or Y
EDG@HLOC	LV SV"1" on page 466	Home location name	8 characters
EDG@HLOT	LV	Home location type	One of AUTO, MANUAL, or blank
EDG@HSKP	LC OPT	Housekeeping functionality switch	ALL, LIMITED
EDG@ID1 - 12	LV	User IDs of authorized users	8 characters each
EDG@INTR	LV SV"1" on page 466	Volume intransit status	Y or N
EDG@IPL	LC	Date check required on IPL flag	Y or N
EDG@IRMM	LV	IRMM use 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
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 LC STATUS	Journal status	8 characters: ENABLED, LOCKED, or DISABLED
EDG@JRNT	LC OPT	Journal transaction	3 characters: NO or YES
EDG@JRNU	LC	Journal percentage used	Numeric: 0 - 100
EDG@KEL1	LV	Key encryption key label 1	1 to 64 characters
EDG@KEL2	LV	Key encryption key label 2	1 to 64 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@KEM1	LV	Key encoding mechanism for key label 1	LABEL or HASH
EDG@KEM2	LV	Key encoding mechanism for key label 2	LABEL or HASH
EDG@KEYF	SD"1" on page 466 SV"1" on page 466	Key from	Character
EDG@KEYT	SD"1" on page 466 SV"1" on page 466	Key to	Character
EDG@LBL	LV SV"1" on page 466	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@LCDT	LB, LD, LO, LP, LR, LS, LV	Last change date	Date format
EDG@LCID	LB, LD, LO, LP, LR, LS, LV	Last change user ID	8 characters. Internal values start with *
EDG@LCSI	LB, LD, LO, LP, LR, LS, LV	Last change system 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@LCTM	LB, LD, LO, LP, LR, LS, LV	Last change time	8 characters (hh:mm:ss)
EDG@LCUD	LB, LD, LO, LP, LR, LS, LV	Last "user" change date	Date format
EDG@LCUT	LB, LD, LO, LP, LR, LS, LV	Last "user" change time	8 characters (hh:mm:ss)
EDG@LDAM	LC	Automove	1 character: Y or N
EDG@LDD	LD	Last used DD name	8 characters
EDG@LDDF	LC"1" on page 466	Existence of a location definition	YES or NO
EDG@LDEV	LD, LV	Last drive	4 characters
EDG@LDLC	LC"3" on page 466	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" on page 466	Location type	One of AUTO, MANUAL, STORE, or blank
EDG@LDMN	LC"3" on page 466	Media name	8 characters
EDG@LDMT	LC"1" on page 466	Management type	BIN, NOBINS, or blank
EDG@LDPR	LC"1" on page 466	Location priority	Numeric: 0 - 9999
EDG@LJOB	LD	Last used job name	8 characters
EDG@LOAN	LV SV"1" on page 466	Volume loan location	8 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
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" on page 466	Location of volume, or bin number	Storage location name
	LR SR"1" on page 466	Location of volume, rack number	SHELF, or an 8 character name of a system-managed library name
	AS LS SS"1" on page 466	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" on page 466	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@LRED	LC OPT, LD	Last reference extra days	Numeric: 0 - 93000
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@MCAT	LC OPT	SMS management class attributes enabled	One of: ALL NONE VRSELXDI
EDG@LVC	LV SV"1" on page 466	Current label version	One of 1,3,4 or blank
EDG@LVN	LV SV"1" on page 466	Required label version	One of 3,4 or blank
EDG@MDNF	LV, LC"17" on page 466	Media information name	8 characters
EDG@MDRA	LC"18" on page 466	MEDINF replace policy for age	Numeric 0-99999
EDG@MDRP	LC"18" on page 466	MEDINF replace policy for permanent errors	Numeric 0-99999

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@MDRT	LC"18" on page 466	MEDINF replace policy for temporary errors	Numeric 0-99999
EDG@MDRW	LC"18" on page 466	MEDINF replace policy for write mount count	Numeric 0-99999
EDG@MDRX	LV, LC"17" on page 466	External recording technology	8 characters
EDG@MDS	LC	Control data set data set name	44 characters
EDG@MDT	LC	Control date set create date	Date format
EDG@MDTX	LV, LC"17" on page 466	External media type	8 characters
EDG@MEDA	LV SV"1" on page 466	Tape special attributes	One of NONE or RDCOMPAT
EDG@MEDC	LV SV"1" on page 466	Tape compaction type	One of *, NONE, or YES
EDG@MEDN	CV LB LC"1" on page 466 LC OPT LR LV SB"1" on page 466 SR"1" on page 466 SV"1" on page 466	Media name	8 characters
EDG@MEDR	LV SV"1" on page 466 LC"17" on page 466	Tape recording technology	8 characters. One of *, 18TRACK, 36TRACK, 128TRACK, 256TRACK, 384TRACK, EFMT1, EFMT2, EEFMT2, EFMT3, EEFMT3, ,EFMT4, EEFMT4
EDG@MEDT	LV SV"1" on page 466 LC"17" on page 466	Tape media type	One of *, CST, EAETC, EATC, EAWTC, ECCST, EETC, EEWTC, EHPCT, ETC, EWTC, EXTC, EXWTC, HPCT
EDG@MFR	LC"1" on page 466	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" on page 466	Moving-in volume	6 characters
EDG@MOP	LC, LC VLPOOL"1" on page 466	Masteroverwrite	One of ADD, MATCH, LAST, or USER
EDG@MOV	LB, SB"1" on page 466	Moving-out volume	6 characters
EDG@MOVM	LV	Move mode	AUTO or MANUAL
EDG@MRP	LC	Maximum retention period	NOLIMIT or Numeric: 0 - 93000
EDG@MSG	LC"2" on page 466	Security classification message option	Y or N

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@MSGF	LC	Case of message text	M or U
EDG@MST	LC"1" on page 466	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" on page 466	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" on page 466	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" on page 466	vital record specification name	8 characters
EDG@NDRP	LC OPT	Default retention period for RM=EXPDT non-GDG data sets	Numeric: 0 -93000 or 11 characters: PERMANENT
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" on page 466 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" on page 466 SS"1" on page 466	Next vital record specification name	8 characters
EDG@NWCT	LC OPT	Default WHILECATALOG setting for RM=EXPDT non-GDG data sets	12 characters: OFF, ON, or UntilExpired
EDG@OAC	LV	Owner access	One of READ, UPDATE, or ALTER
EDG@OBMN	LV	Old bin number media name	8 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
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@OLON	LV	Old loan location	8 characters
EDG@OLOT	LV	Old location type	One of AUTO, MANUAL, STORE, INCTNR, or blank
EDG@OPL	LC"1" on page 466	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@ORIA	LC OPENRULE"1" on page 466	Input action	ACCEPT, REJECT, or IGNORE
EDG@ORII	LC OPENRULE"1" on page 466	Input IGNORE condition (BY)	ANY, NONSPECIFIC, SPECIFIC, or blank
EDG@ORIR	LC OPENRULE"1" on page 466	Input REJECT condition (BY)	SYSID, CATLG, 'SYSID,CATLG', or blank
EDG@ORIS	LC OPENRULE	Input IGNORE SAFRC8	BYPASSAFRC8,CHECKSAFRC8
EDG@OROA	LC OPENRULE"1" on page 466	Output action	ACCEPT, REJECT, or IGNORE
EDG@OROI	LC OPENRULE"1" on page 466	Output IGNORE condition (BY)	ANY, NONSPECIFIC, SPECIFIC, or blank
EDG@OROR	LC OPENRULE"1" on page 466	Output REJECT condition (BY)	SYSID, CATLG, 'SYSID,CATLG', or blank
EDG@OROS	LC OPENRULE	Output IGNORE SAFRC8	BYPASSAFRC8,CHECKSAFRC8
EDG@ORTP	LC OPENRULE"1" on page 466	Type of open rule entry	RMM or NORMM
EDG@ORVE	LC OPENRULE"1" on page 466	Volume range end or blank if volume specified	6 characters
EDG@ORVL	LC OPENRULE"1" on page 466	Volume serial number, specific or generic, or blank if volume range is specified	6 characters
EDG@ORVS	LC OPENRULE"1" on page 466	Volume range start or blank if volume specified	6 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@OVL	LC"1" on page 466	Position of the volume serial number in mount message	Numeric: 0 - 999
EDG@OVOL	LB, SB"1" on page 466	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" on page 466	Owner of volume on which data set resides	Character 8
	SP"1" on page 466	Software product owner	Character 8
	SV"1" on page 466	Volume owner	Character 8
	SS"1" on page 466	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" on page 466	Pool description	40 characters
EDG@PDSC	LP	Software product description	30 characters
EDG@PEND	LV SV"1" on page 466	Actions pending for volume E=ERASE I=INIT N=NOTIFY O=OWNER R=REPLACE S=SCRATCH	One of I, O, OE, OEN, OI, OIE, OIEN, OIN, ON, R, RE, REN, RI, RIE, RIEN, RIN, RN, S, SE, SEN, SI, SIE, SIEN, SIN, SN
EDG@PID	LC, LR, SR"1" on page 466	Pool prefix	6 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@PLN	LC"1" on page 466	Pool name	8 characters
EDG@PNME	LP SP"1" on page 466	Software product name	30 characters
EDG@PNUM	LP LV SP"1" on page 466	Software product number	8 characters
EDG@PRD	LV	Number of permanent read errors	Numeric: 0 - 99999
EDG@PRF	LC"1" on page 466	Pool definition RACF option	Y or N
EDG@PRTY	LS SS"1" on page 466	Priority	Numeric: 0 - 9999
EDG@PSFX	LC	Parmlib member suffix	2 characters
EDG@PSF2	LC OPT	2nd parmlib member suffix	2 characters
EDG@PSN	LC"1" on page 466	Pool definition system ID	8 characters
EDG@PSZ6	LD LV	Physical space used	Factored number FF*num*. Up to 12 characters: FF is KB, MB, GB, TB *num* is numeric 0 - 9999999999
EDG@PTNA	LC PRTITION"1" on page 466	NOSMT Action for partition entry	ACCEPT or IGNORE
EDG@PTNL	LC PRTITION"1" on page 466	Location name for NOSMT	SHELF or LOCDEF defined home storage location name, or blank
EDG@PTP	LC"1" on page 466	Pool definition pool type	R or S
EDG@PTSA	LC PRTITION"1" on page 466	SMT Action for partition entry	ACCEPT or IGNORE
EDG@PTTP	LC PRTITION"1" on page 466	Type of partition entry	RMM or NORMM
EDG@PTVE	LC PRTITION"1" on page 466	Volume range end, or blank if volume is specified	6 characters
EDG@PTVL	LC PRTITION"1" on page 466	Volume serial number, specific or generic, or blank if volume range is specified	6 characters
EDG@PTVS	LC PRTITION"1" on page 466	Volume range start, or blank if volume is specified	6 characters
EDG@PVL	LV	Previous volume in sequence	6 characters
EDG@PWT	LV	Number of permanent write errors	Numeric: 0 - 99999

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@RBN	LC	Number of bin numbers in REMOTE storage location	Numeric: 0 - 999999
EDG@RBYS	LV SV"1" on page 466	Retained by set	3 characters: YES or NO
EDG@RC	All subcommands"7" on page 466	Reason code	Numeric
EDG@RCF	LC	Installation RACF support	One of N, P, A, or C
EDG@RCFM	LD	Data set record format (RECFM)	4 characters
EDG@RCK	AB AR AV"4" on page 466 CV"5" on page 466 DB DR LP"1" on page 466 LB LR LV SB"1" on page 466 SR"1" on page 466 SV"1" on page 466	Rack number	6 characters
EDG@RDT	LC	Date of last control data set report extract	Date format
EDG@RET	LS"11" on page 466	Retention type	One of BYDAYC, CYCLES, DAYS, REFDAYS, VOLUMES, or XTRADAYS
	SS"1" on page 466	Retention type	One of BYDAYC, CYCLES, DAYS, REFDAYS, VOLUMES, XTRADAYS, or blank
EDG@RLPR	LV	Required location priority	1 - 9999, or blank
EDG@RM	LC OPT	Retention method	EXPDT
	LV VOL		VRSEL
EDG@RMID	LC CNTL	Started procedure name	Up to 17 characters. One of: • procedure name • job name • concatenation of procedure name.identifier

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@RMSB	LV VOL	Retention method set by	Up to 10 characters. One of: CMD CMD_DEF CNVT DEFTABLE EXPORT_DEF INERS_DEF LASTREF LCS_DEF MC_ATTR OCE_DEF OCE_EXIT UNDEFINED
EDG@RST	LB LR SB"1" on page 466 SR"1" on page 466	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	SD"1" on page 466 SV"1" on page 466	Retention date	Julian date format
EDG@RTDT	LD LV SD"1" on page 466 SV"1" on page 466	Retention date	Calendar date, CATRETPD, 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" on page 466	Retain while cataloged	YES or NO
	SS"1" on page 466		YES, NO, or blank
EDG@SC	LD	Storage class name	8 characters
EDG@SC1	LS SS"1" on page 466	Vital record specification first storage location days or cycles or volumes	Numeric: 1 - 99999
EDG@SCRM	LC VLPOOL"1" on page 466	Scratch mode	6 characters: AUTO or MANUAL
EDG@SDT	LV	Movement tracking date	Date format
EDG@SEC	LC"2" on page 466	Security classification number	Numeric: 0 - 255
EDG@SEQ	LV	Volume sequence number	Numeric: 1 - 9999
EDG@SG	LD	Storage group name	8 characters
EDG@SGN	LV	Storage group name	8 characters
EDG@SID	LC	SMF system ID	8 characters

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@SLM	LC	MAXHOLD value	Numeric: 10 - 500
EDG@SMF	LC"2" on page 466	Security classification SMF flag	Y or N
EDG@SMI	LC"1" on page 466	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@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	45 characters: 1 to 45 numeric characters including colon, 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 type	Numeric: 0, 42, 128 - 255. 0 is no security records written; 42, 128 - 255 are the security record types
EDG@STDS	LC STATUS	Debug setting	8 characters: DISABLED, OCE+SNAP, OCE, SNAP

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@STEP	LD	Step name in job that created the data set	8 characters
EDG@STIS	LC STATUS	IP verb state	1 character: blank or **<** request has started **>** request has ended
EDG@STIT	LC STATUS	IP verb time	8 characters: *hh:mm:ss*
EDG@STIV	LC STATUS	IP verb	5 characters: blank, READ, WRITE, CONN, CLOSE
EDG@STLA	LC STATUS	Local active tasks	Numeric: 0 - 999
EDG@STLH	LC STATUS	Local held tasks	Numeric: 0 - 999
EDG@STLO	LC STATUS	Local tasks	Numeric: 0 - 999
EDG@STLR	LC STATUS	Last CDS reserve time	8 characters: *hh:mm:ss*
EDG@STNH	LC STATUS	New requests held	7 characters: NOTHELD, HELD
EDG@STPL	LC STATUS	PDA Trace levels	7 characters: A string 1,2,3,4 where one or more of the values may be blank. If a number is set it indicates the trace level is on.
EDG@STQC	LC STATUS	Queued catalog requests	Numeric: 0 - 999999
EDG@STQN	LC STATUS	Queued nowait requests	Numeric: 0 - 999999
EDG@STQR	LC STATUS	Queued requests	Numeric: 0 - 999999
EDG@STRF	LC STATUS	Task requested function	5 characters: ACS, ADD, API, BKUP, C/S, CAT, CHG, CLOSE, DEL, DFHSM, HSKP, INERS, LC, LCSUX, LIST, LS, OPEN, SR, VRS, WTO Any other values are internal to DFSMSrmm.
EDG@STRH	LC STATUS	CDS Reserved	5 characters: + ENQ, - DEQ
EDG@STRM	LC STATUS	DFSMSrmm status	8 characters: ACTIVE, RESET, QUIESCED
EDG@STRT	LC STATUS	Task requestor's system ID	8 characters
EDG@STSA	LC STATUS	Server active tasks	Numeric: 0 - 999
EDG@STSH	LC STATUS	Server held tasks	Numeric: 0 - 999

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@STSL	LC STATUS	Server listener	8 characters: blank, INACTIVE, or ACTIVE
EDG@STSO	LC STATUS	Server tasks	Numeric: 0 - 999
EDG@STST	LC STATUS	Task start time	8 characters: *hh:mm:ss*
EDG@STTQ	LC STATUS	Task requestor's ID	
EDG@STTR	LC STATUS	Task requestor type	3 characters: JOB, STC, TSU
EDG@STTS	LC STATUS	Task status	1 character: blank or **H** subject to HOLD **C** subject to CANCEL **+** This task holds the RESERVE on the DFSMSrmm CDS
EDG@STTT	LC STATUS	Task token	8 alphanumeric characters
EDG@STVC	LV, SV"1" on page 466	Count of stacked volumes	Numeric: 10 digits maximum
EDG@SUR	LO	Owner's surname	20 characters
EDG@SYS	LV, SV"1" on page 466	Creation system ID	1-to-8 characters
EDG@TAC	LC"1" on page 466	Reject prefix type	One of READONLY or NONE
EDG@TLR	LS, SS"1" on page 466	Time last referenced	8 characters (hh:mm:ss)
EDG@TRD	LV	Number of temporary read errors	Numeric: 0 - 99999
EDG@TVXD	LC	TVEXTPURGE days	0 to nnnn
EDG@TVXP	LC	Tape volume exit purge option	One of RELEASE, EXPIRE, or NONE
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" on page 466		One of GDG, PGDG, DSN, VOL, or NAME
EDG@TYPF	SD"1" on page 466 SV"1" on page 466	Key from	Character
EDG@TYPT	SD"1" on page 466 SV"1" on page 466	Key from	Character
EDG@UDT	LC	Last update date	Date format

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@UEX	LS"8" on page 466	Retain until expired	YES or NO
	SS"1" on page 466		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: 1 - 9999999999. -1 indicates that USE6 should be used.
EDG@USE6	LV	Volume usage (Application written data)	Factored number FF*num*. Up to 12 characters: FF is KB, MB, GB, TB *num* is numeric 0 - 9999999999
EDG@UTC	LC CNTL	Common time	One of ENABLED or DISABLED
EDG@UTM	LC	Last update time	8 characters (hh:mm:ss)
EDG@VAC	LV	Volume access	One of READ, UPDATE, or NONE
EDG@VACT	LC	VRSMIN action	One of FAIL, INFO, WARN, or OFF
EDG@VANX	LS SS"1" on page 466	Next vital record specification type	One of NEXT, AND, or blank
EDG@VCAP	LV, LC"17" on page 466	Volume/media capacity	Numeric: 0 - 2147483647
EDG@VCHG	LC	VRSCHANGE value	One of INFO, VERIFY
EDG@VDD	LS	Vital record specification delay days"11" on page 466	Numeric: 0 - 99
EDG@VDRA	LC	VRSDROP action	One of FAIL, INFO, OFF, or WARN
EDG@VDRC	LC	VRSDROP count	Numeric: 0-2,147,483,647
EDG@VDRP	LC	VRSDROP percent	Numeric: 0-100
EDG@VDT	LC	Date of last inventory management vital record processing	Date format
EDG@VER	LP LV SP"1" on page 466	Software product version	vvrrmm
EDG@VEX	LD	VRSEL exclude	YES or NO
EDG@VJBN	LD	VRS job name mask	8 characters
	LS"8" on page 466	Job name	8 characters
	SS"1" on page 466	Job name	8 characters"12" on page 466

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@VLN	LO	Number of owned volumes	Numeric: 0 - 4294867296
	LP SP"1" on page 466	Number of software product volumes	Numeric: 0 - 999
EDG@VM	LV	VM use flag	Y or N
EDG@VMIN	LC	VRSMIN count value	A number
EDG@VMV	LD	Vital record specification management value	8 characters, defined by your installation
EDG@VNDR	LV VOL	Vendor information	8 alphanumeric or special characters
EDG@VNME	LD	Vital record specification name	44 characters
EDG@VOL	AV GV LD LP"1" on page 466 LB LR LV LS"10" on page 466 SB"1" on page 466 SD"1" on page 466 SP"1" on page 466 SR"1" on page 466 SV"1" on page 466	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" on page 466 SS"1" on page 466	Vital record count	Numeric: 1 - 99999
EDG@VREA	LC	VRSRETAIN action	One of FAIL, INFO, OFF, or WARN
EDG@VREC	LC	VRSRETAIN count	Numeric: 0-2,147,483,647
EDG@VREP	LC	VRSRETAIN percent	Numeric: 0-100
EDG@VRJ	LC	Vital record specification job name	Numeric: 1 or 2
EDG@VRS	SS"1" on page 466	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" on page 466	SCRATCHIMMEDIATE release option	YES or NO
EDG@VRSL	LC OPTION	VRSEL value	NEW
EDG@VRSR	LD SD"1" on page 466	Data set vital record status	YES or NO
EDG@VRXI	LV LS SS"1" on page 466	EXPIRYDATEIGNORE release option	YES or NO

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@VSCD	LD	Primary vital record specification subchain date	Date format
EDG@VSCN	LD	Primary vital record specification subchain name	8 characters
EDG@VST	LV SV[“1” on page 466]	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	Volume write mount count	Numeric: 0 - 99999
EDG@WCTL	LD	WHILECATALOG setting for data set	12 characters: OFF, ON, or UntilExpired
EDG@WORM	LV	Volume is Write Once Read Many (WORM)	YES or NO
EDG@WWID	LV	World wide identifier	24 hexadecimal characters
EDG@XDC	LC[“1” on page 466]	Expiration date check	One of Y, N, or O
EDG@XDRA	LC	EXPDTDROP action	One of FAIL, INFO, OFF, or WARN
EDG@XDRC	LC	EXPDTDROP count	Numeric: 0-2,147,483,647
EDG@XDRP	LC	EXPDTDROP percent	Numeric: 0-100

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@XDSB	LV VOL, LD	Expiration date set by	12 characters: blank (not set) CATLG_DAYS CATRETPD CMD CMD_DEF CMD_VOLCAT CNVT DEFTABLE EXPORT LASTREF LCS LCS_DEF OCE_DEF OCE_EXIT OCE_JFCB OCE_MAX OCE_MC OCE_VOLCAT TVEXTPURGE
EDG@XDT	LC	Date of last inventory management expiration processing	Date format
	LD SD"1" on page 466	Data set expiration date	Date format
	LV SV"1" on page 466	Volume expiration date	Date format
EDG@XDTJ	SD SV"1" on page 466	Volume expiration date	Julian date format
EDG@XTM	LC	Time of last inventory management expiration processing	8 characters (hh:mm:ss)
	LD LV	Expiration time	8 characters (hh:mm:ss)
EDG@X100	LC	EDG_EXIT100 installation exit status	8 characters: ENABLED, DISABLED, or NONE
EDG@X200	LC	EDG_EXIT200 installation exit status	8 characters: ENABLED, DISABLED, or NONE
EDG@X300	LC	EDG_EXIT300 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

Table 40. TSO subcommand variables by name (continued)

Variable Name	Abbrev Subcommands	Contents	Format
EDG@2SCN	LD	Secondary vital record specification subchain name	8 characters

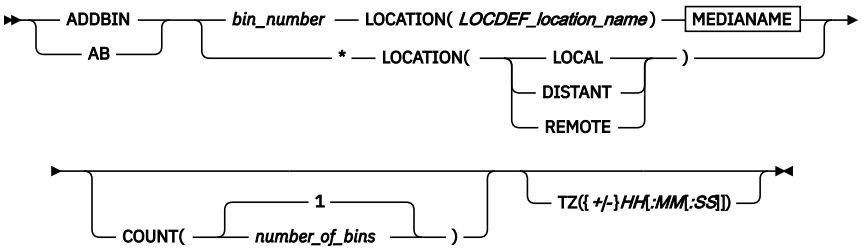
Notes:

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 "Double stem variables" on page 432.
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.
17. The variable is a stem variable when you specify LISTCONTROL with MEDINF operand.
18. Values are always returned and contain the built-in value if REPLACE operand was not specified.

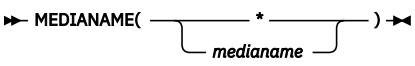
Appendix A. DFSMSrmm TSO subcommands

The IBM tape management product DFSMSrmm provides the RMM TSO subcommand and a set of subcommands to request DFSMSrmm functions.

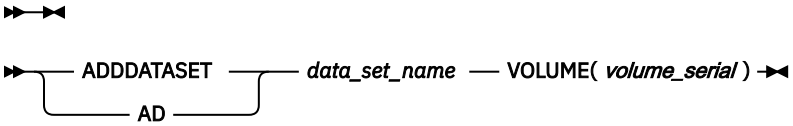
ADDBIN: Adding a bin number in a storage location



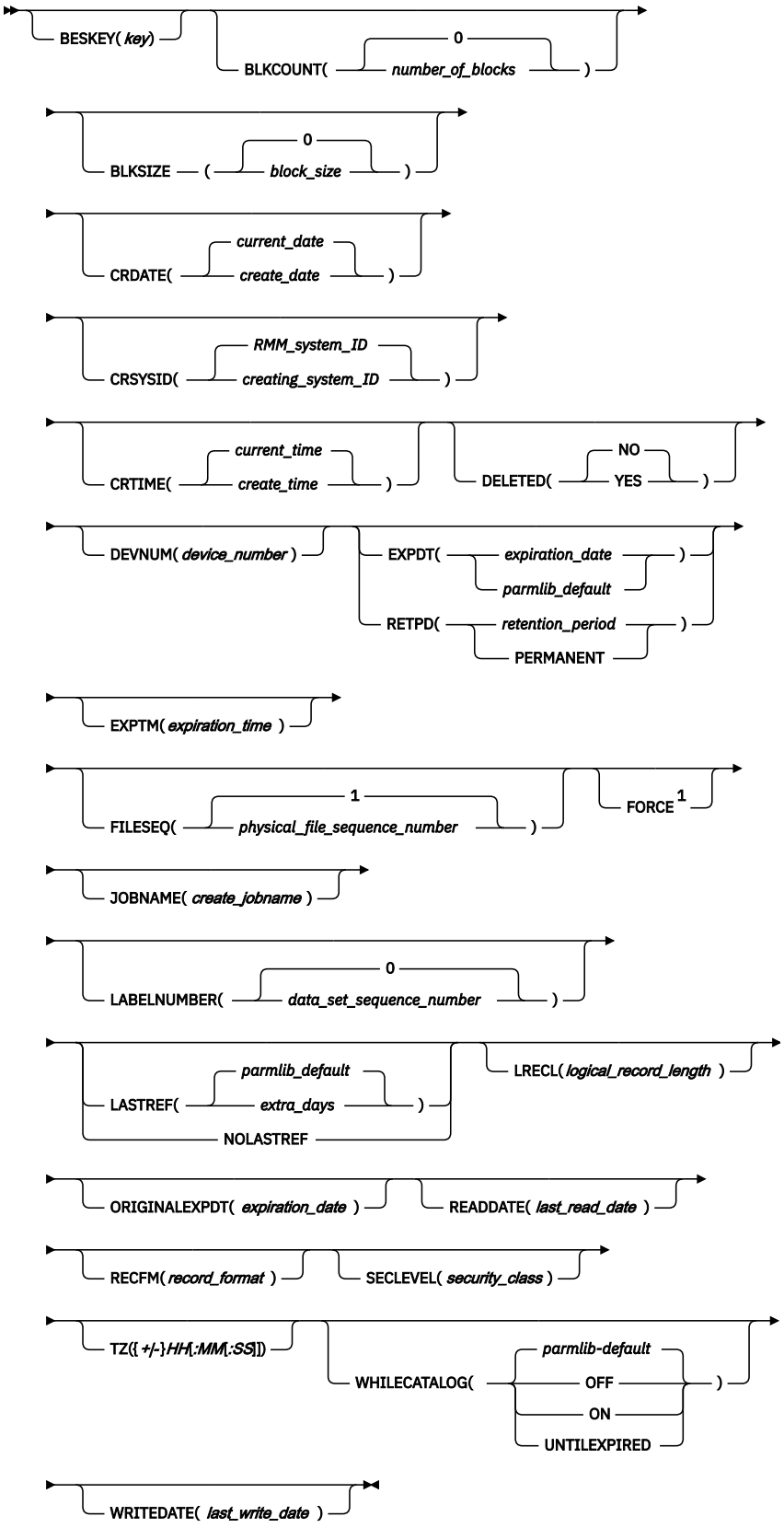
MEDIANAME



ADDDATASET: Adding data set information



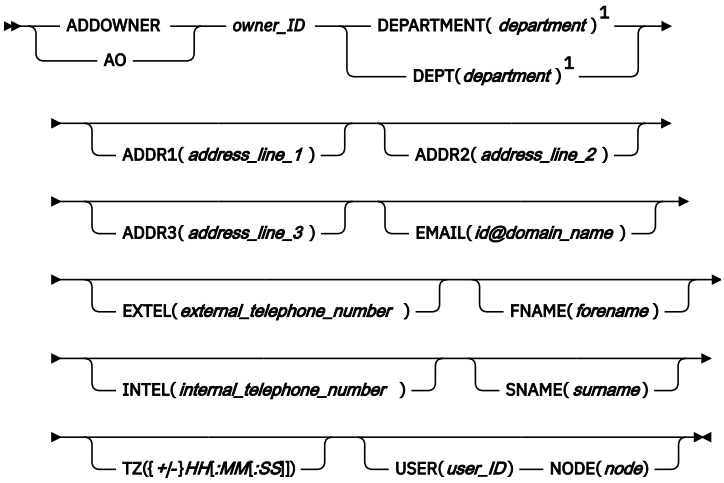
© Copyright IBM Corp. 1992, 2022



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,

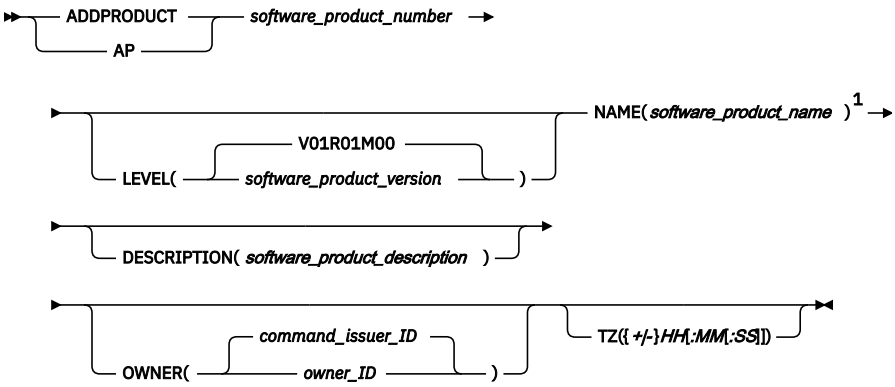
ADDOWNER: Adding owner information



Notes:

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

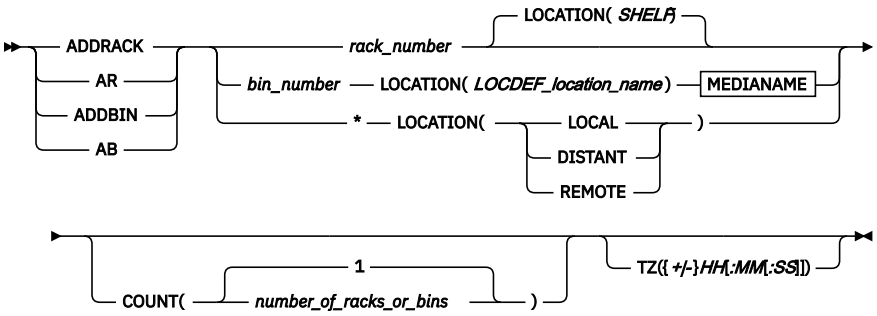
ADDPRODUCT: Adding software product information



Notes:

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

ADDRACK: Adding a shelf location



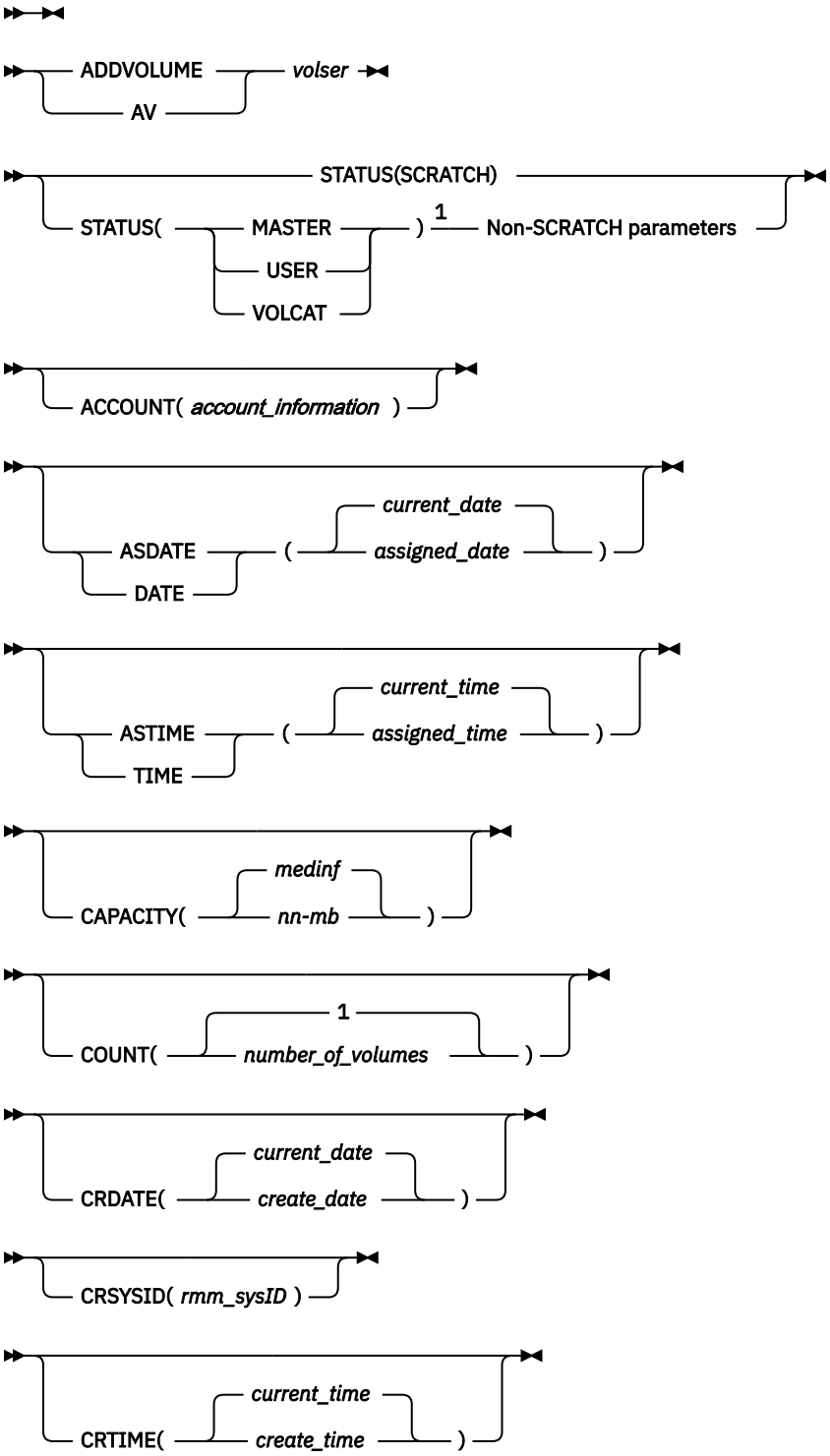
MEDIANAME

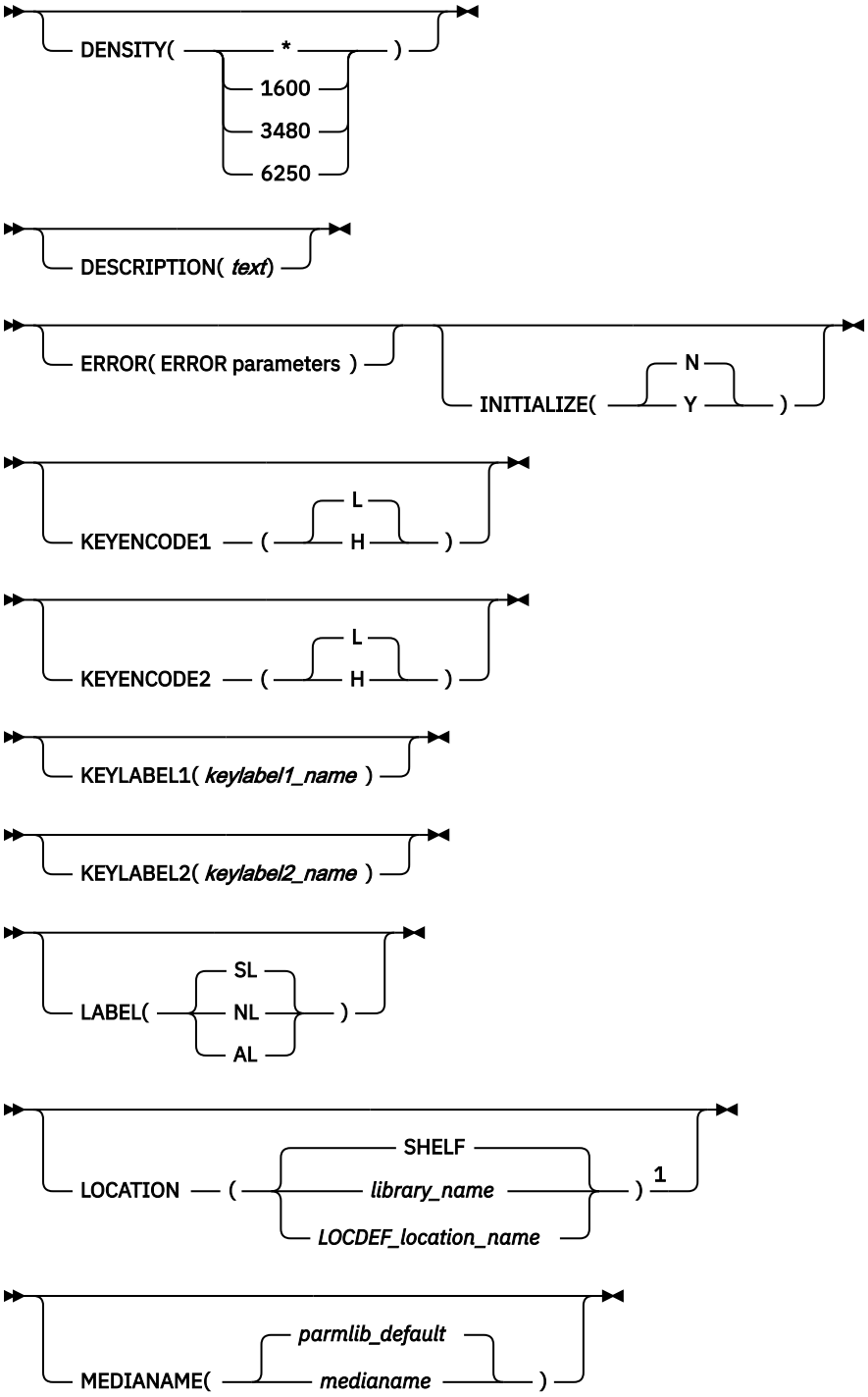
ADDVOLUME: Adding volume information

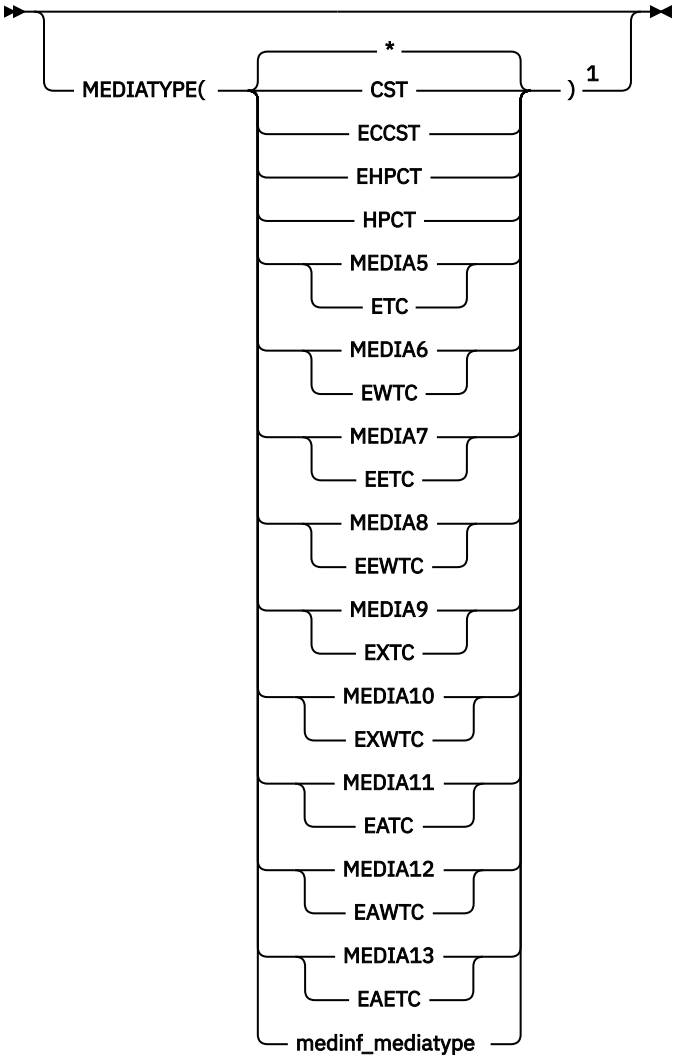
The ADDVOLUME operands are broken down into multiple syntax diagrams:

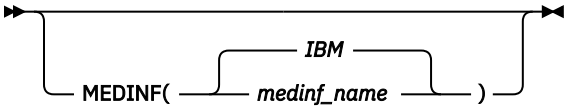
- ADDVOLUME: Volume Operands
- ADDVOLUME: Non-SCRATCH Volume Optional Operands

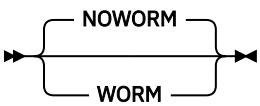
ADDVOLUME: Volume operands

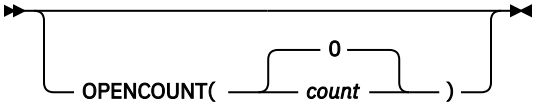


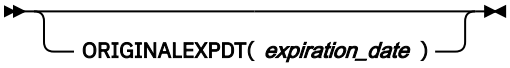

DENSITY(* 1600 3480 6250)
DESCRIPTION(text)
ERROR(ERROR parameters)
INITIALIZE(N Y)
KEYENCODE1 (L H)
KEYENCODE2 (L H)
KEYLABEL1(keylabel1_name)
KEYLABEL2(keylabel2_name)
LABEL(SL NL AL)
LOCATION (SHELF library_name LOCDEF_location_name) 1
MEDIANAME(parmlib_default medianame)

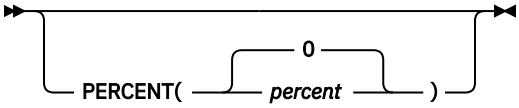

MEDIATYPE(
*
CST
ECCST
EHPCT
HPCT
MEDIA5
ETC
MEDIA6
EWTC
MEDIA7
EEETC
MEDIA8
EEWTC
MEDIA9
EXTC
MEDIA10
EXWTC
MEDIA11
EATC
MEDIA12
EAWTC
MEDIA13
EAETC
medinf_mediatype
) 1

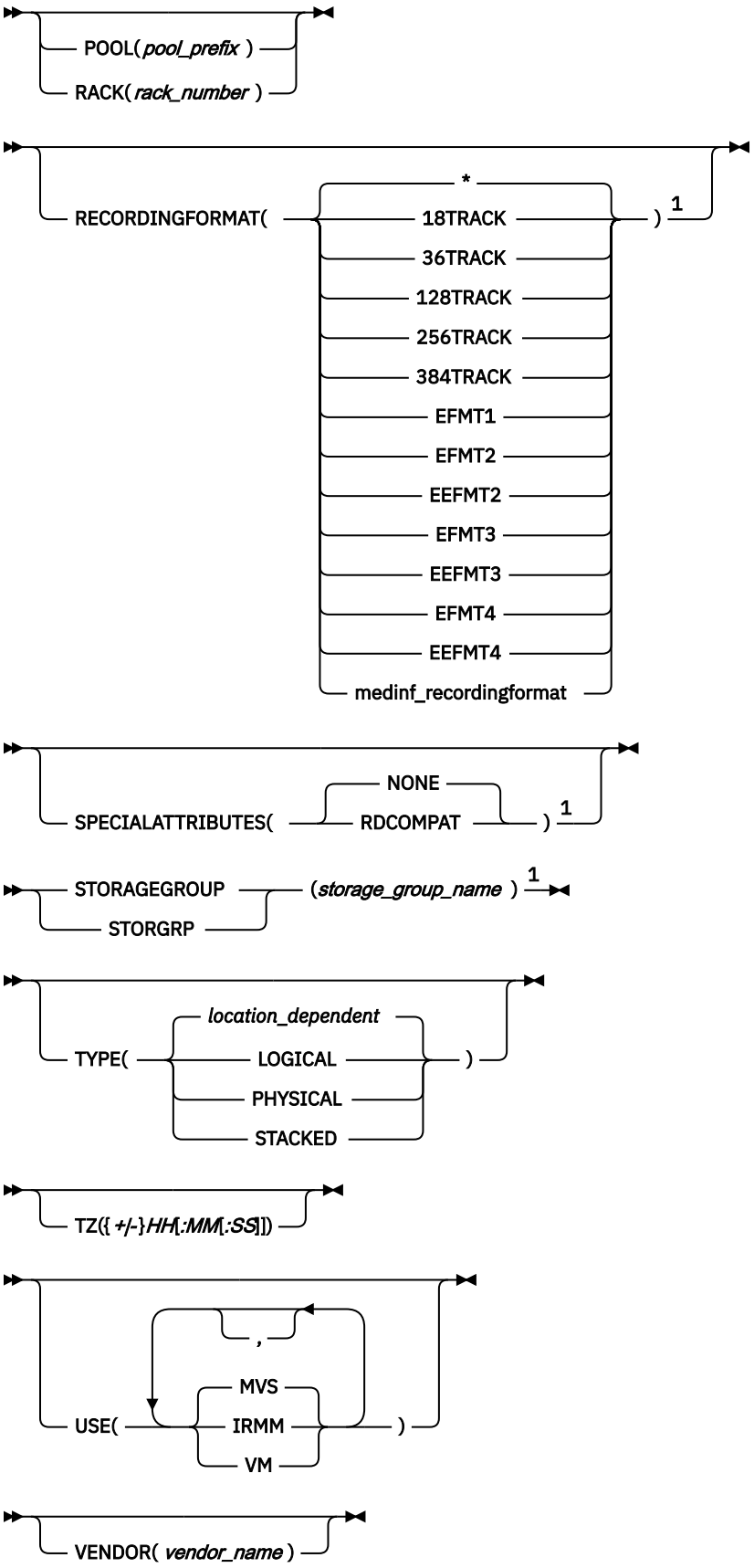

MEDINF(
IBM
medinf_name
)

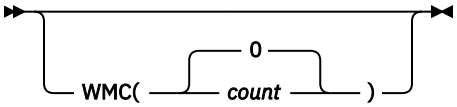

NOWORM
WORM

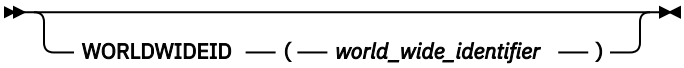

OPENCOUNT(
0
count
)


ORIGINALEXPDT(expiration_date)

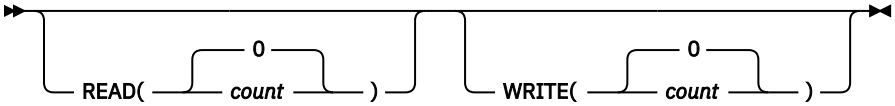

PERCENT(
0
percent
)


POOL(pool_prefix)
RACK(rack_number)
RECORDINGFORMAT(
*
18TRACK
36TRACK
128TRACK
256TRACK
384TRACK
EFMT1
EFMT2
EEFMT2
EFMT3
EEFMT3
EFMT4
EEFMT4
medinf_recordingformat
) 1
SPECIALATTRIBUTES(
NONE
RDCOMPAT
) 1
STORAGEGROUP
STORGRP
(storage_group_name) 1
TYPE(
location_dependent
LOGICAL
PHYSICAL
STACKED
)
TZ({+/-}HH[:MM[:SS]])
USE(
,
MVS
IRMM
VM
)
VENDOR(vendor_name)





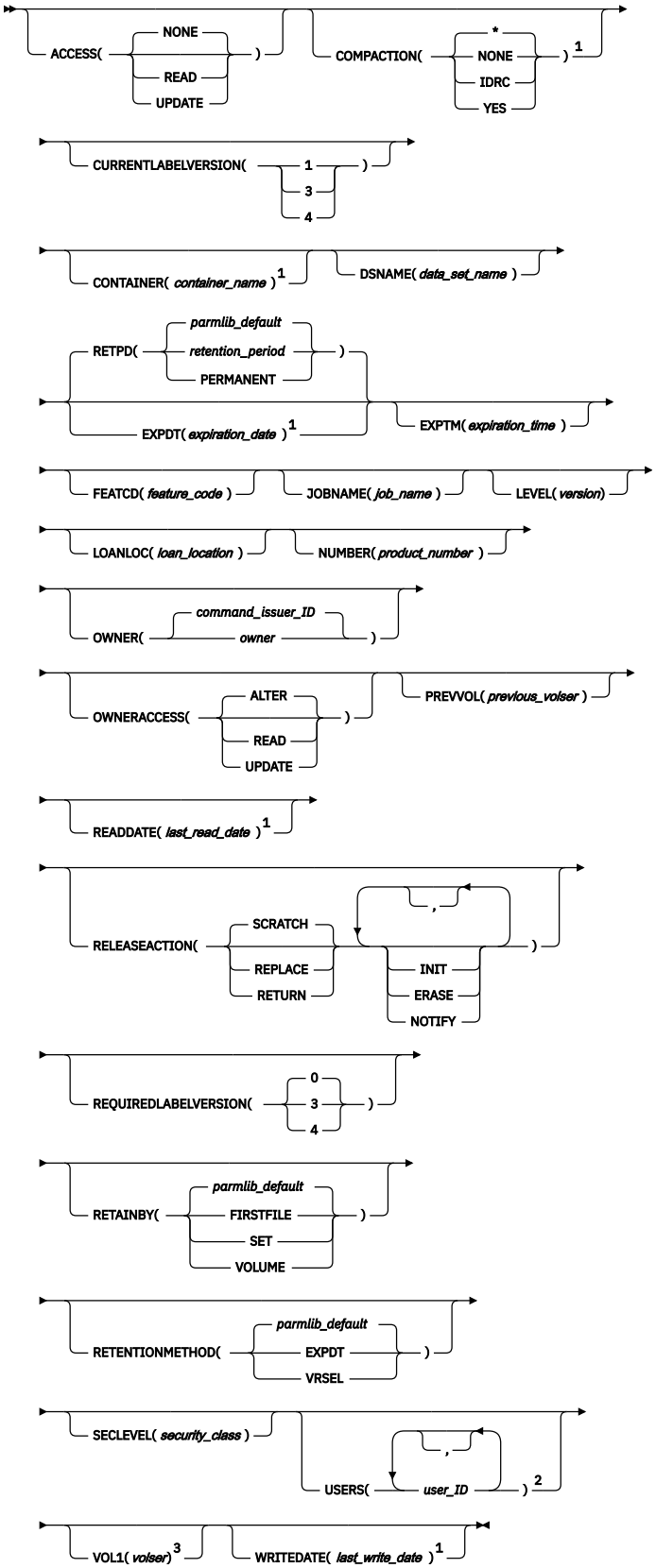
ERROR parameters



Notes:

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

ADDVOLUME: Non-SCRATCH volume optional operands



Notes:

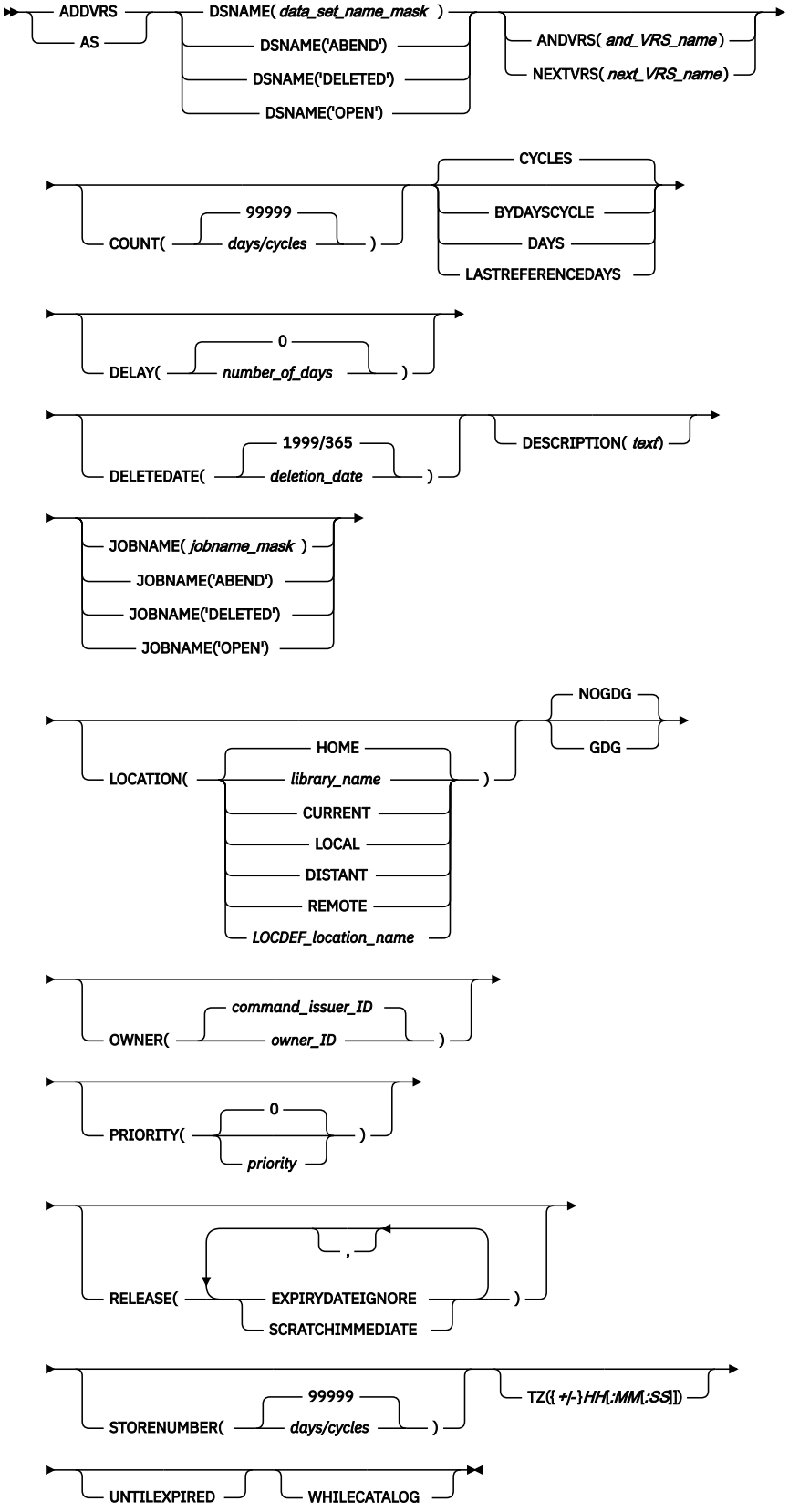
[1] This operand uses the tape configuration database information when STATUS(VOLCAT) is specified.

[2] You can specify a maximum of 12 user IDs.

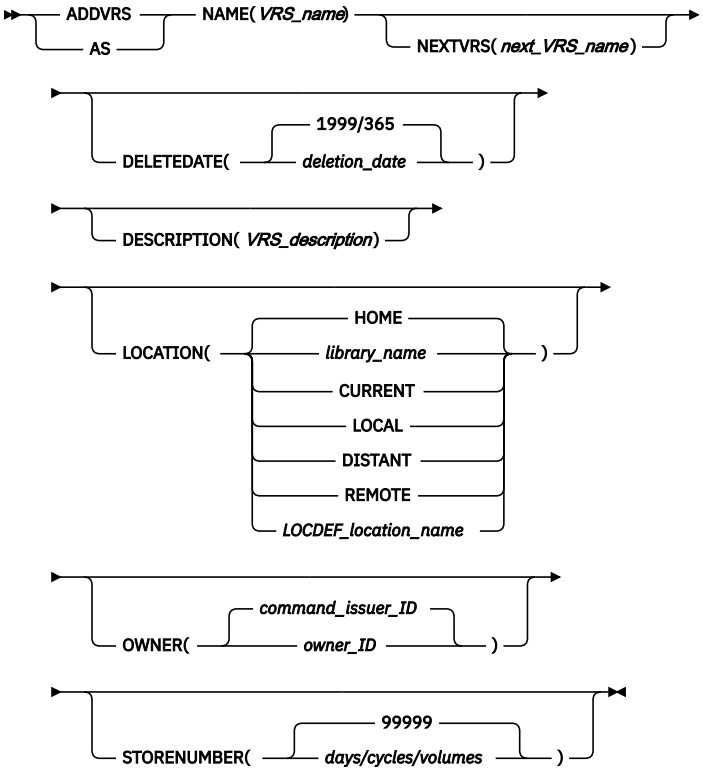
[3] This operand cannot be specified for SCRATCH volumes, LOGICAL volumes, STACKED volumes, or NOLABEL volumes.

ADDVRS: Adding a vital record specification

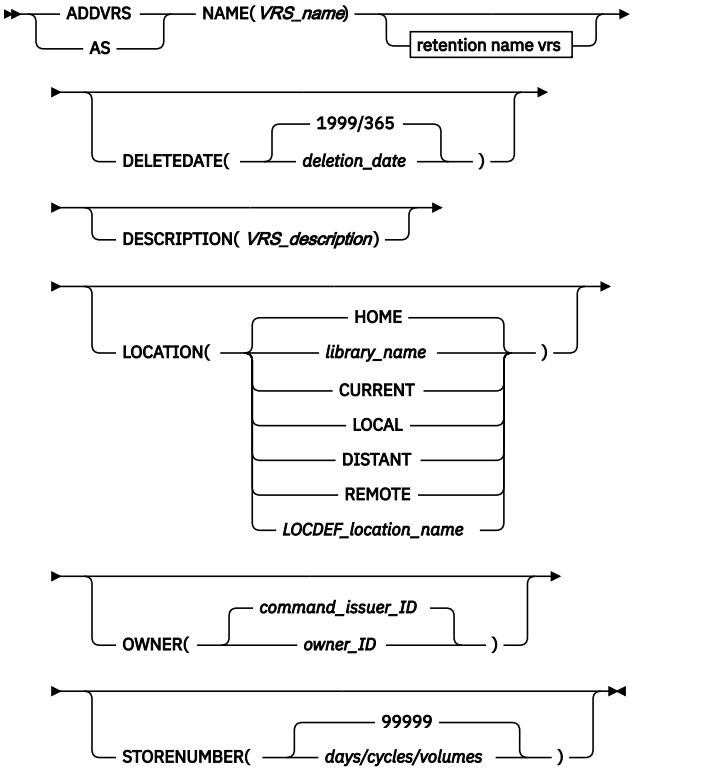
ADDVRS: Adding a data set vital record specification



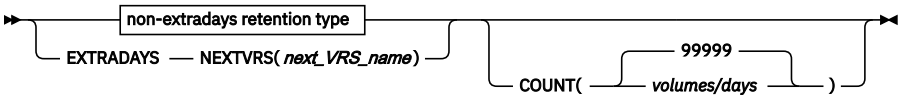
ADDVRS: Adding a location name vital record specification



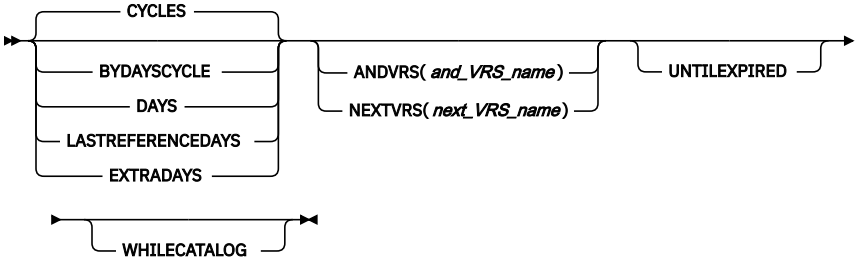
ADDVRS: Adding a retention name vital record specification



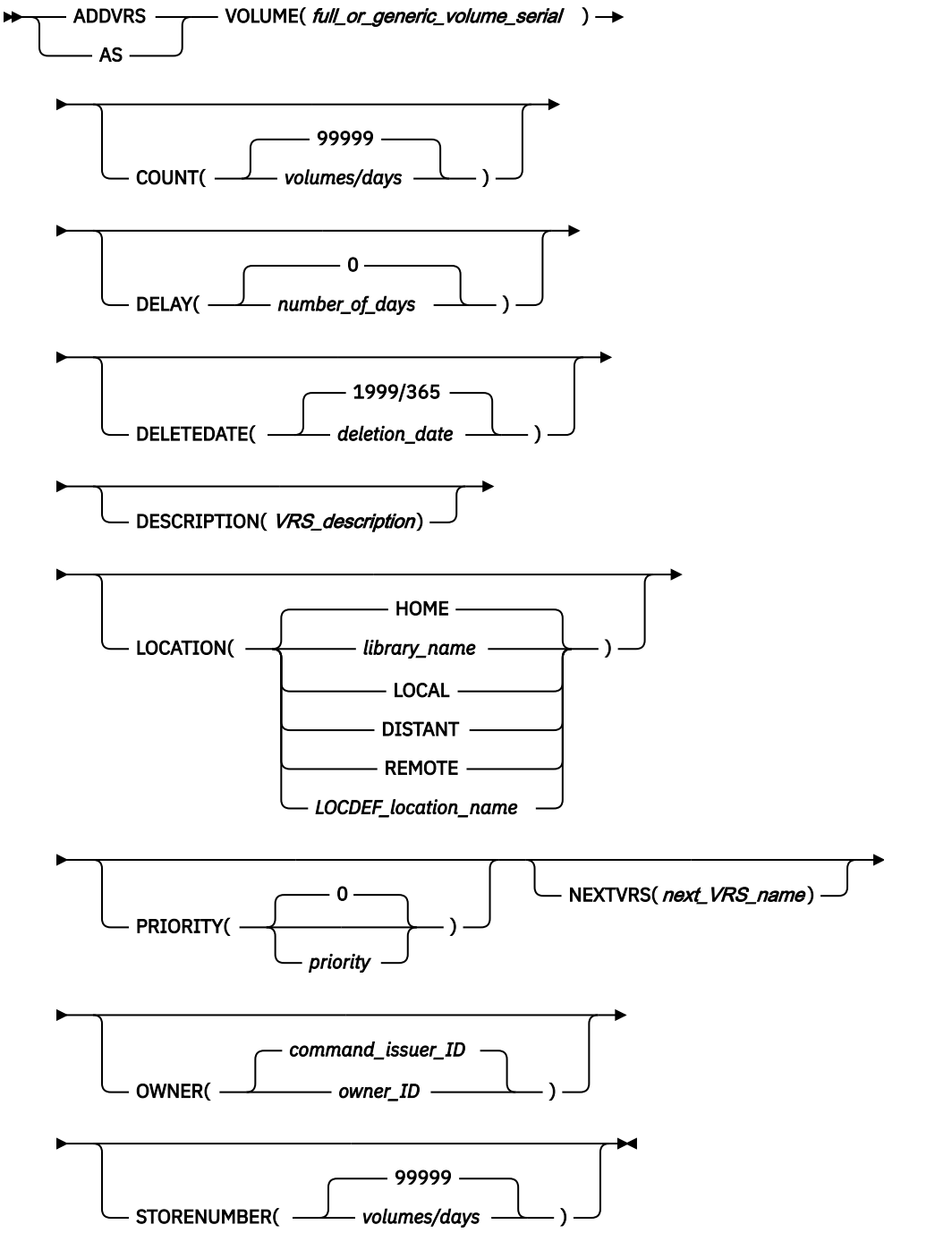
retention name vrs



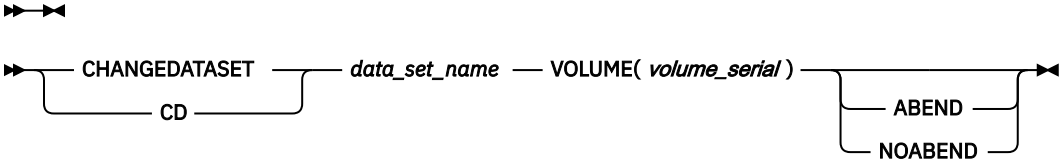
non-extradays retention type

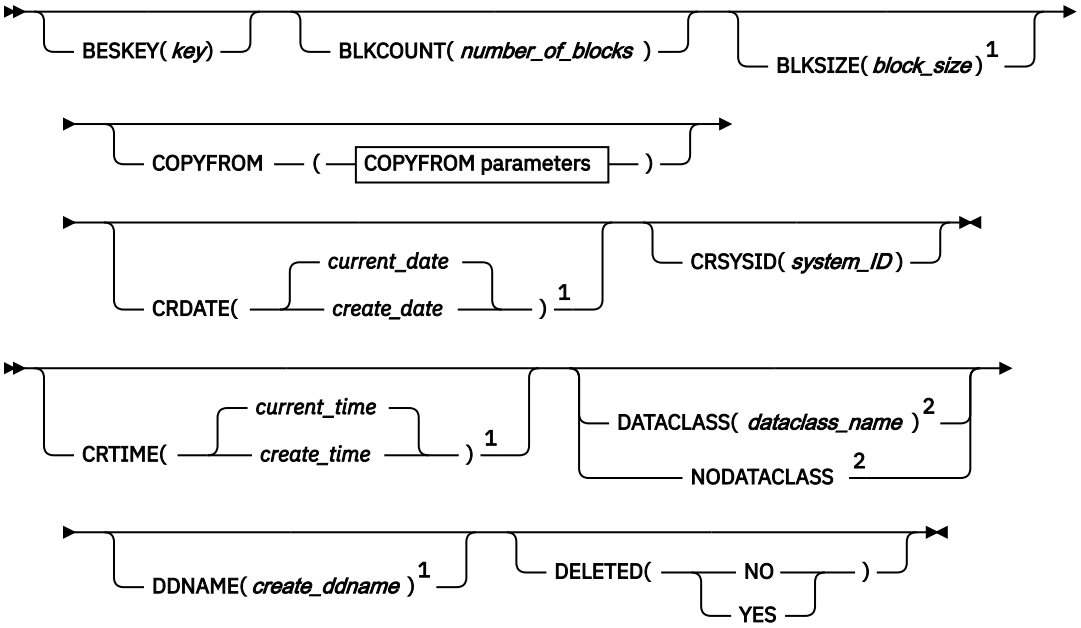


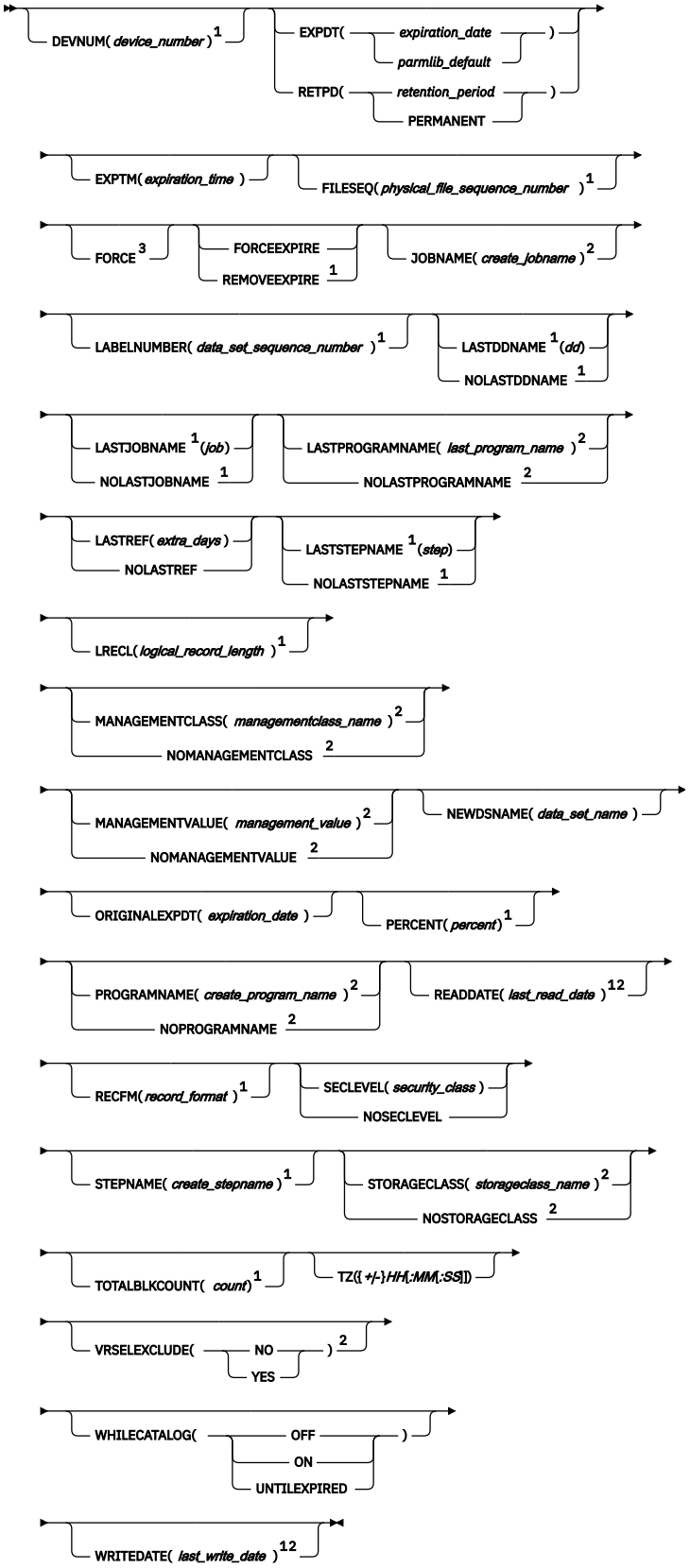
ADDVRS: Adding a volume vital record specification



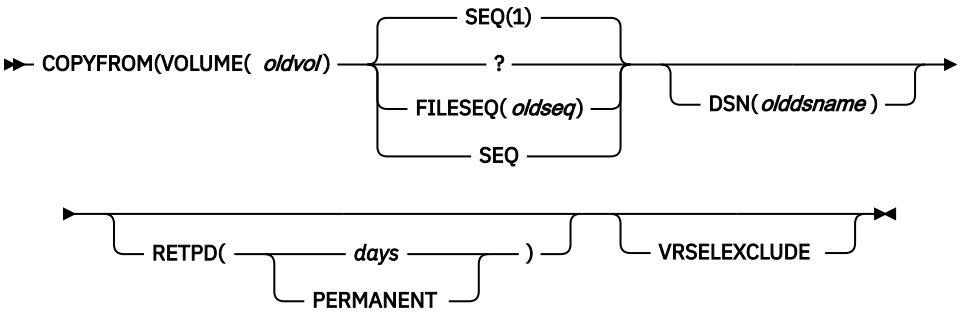
CHANGEDATASET: Changing data set information




BESKEY(key)
BLKCOUNT(number_of_blocks)
BLKSIZE(block_size) 1
COPYFROM (COPYFROM parameters)
CRDATE(current_date create_date) 1
CRSYSID(system_ID)
CRTIME(current_time create_time) 1
DATACLASS(dataclass_name) 2
NODATACLASS 2
DDNAME(create_ddname) 1
DELETED(NO YES)



COPYFROM parameters



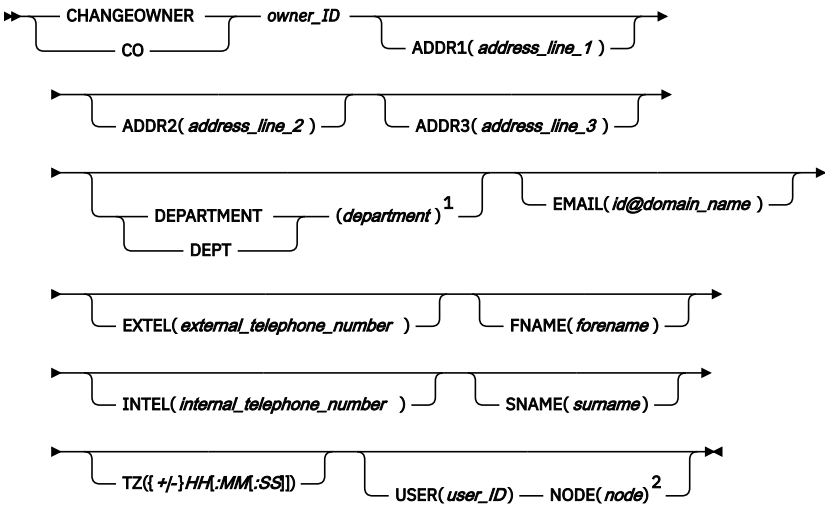
Notes:

[1] When this operand is specified, the FORCE operand must also be specified if DFSMSrmm has automatically recorded information about the volume during O/C/EOV processing of a data set on the volume. 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.

[2] This operand can be specified only 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

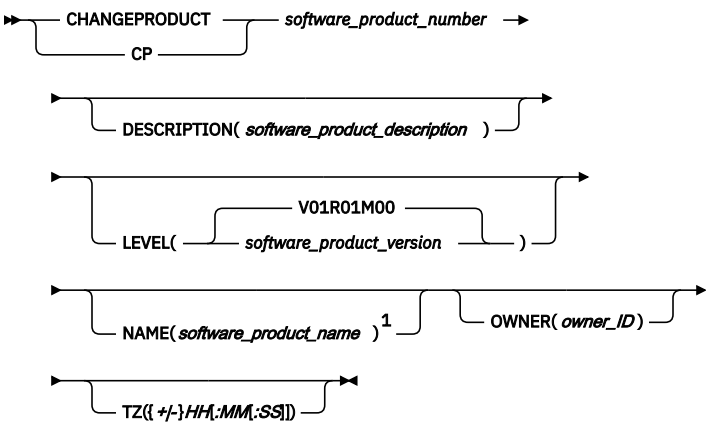


Notes:

[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:

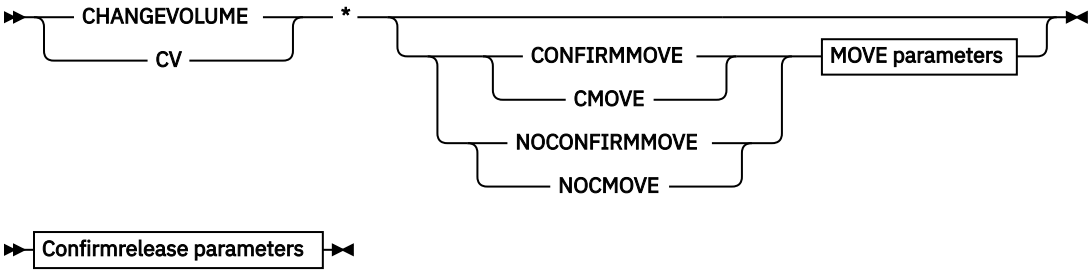
[1] The NAME operand must contain at least one non-blank character.

CHANGEVOLUME: Changing volume information

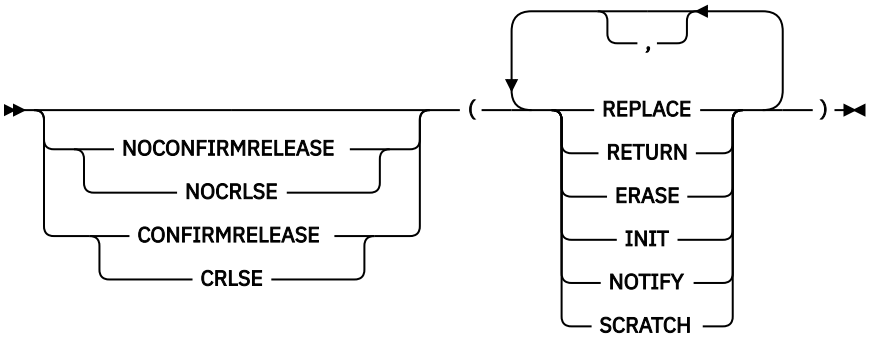
The CHANGEVOLUME operands are broken down into multiple syntax diagrams:

- CHANGEVOLUME: Confirmrelease and global movement operands
- CHANGEVOLUME: Specific volume optional operands, which spans multiple pages.

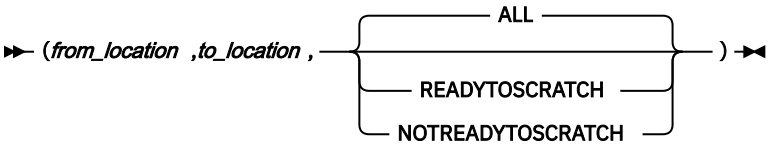
CHANGEVOLUME: Confirmrelease and global movement operands



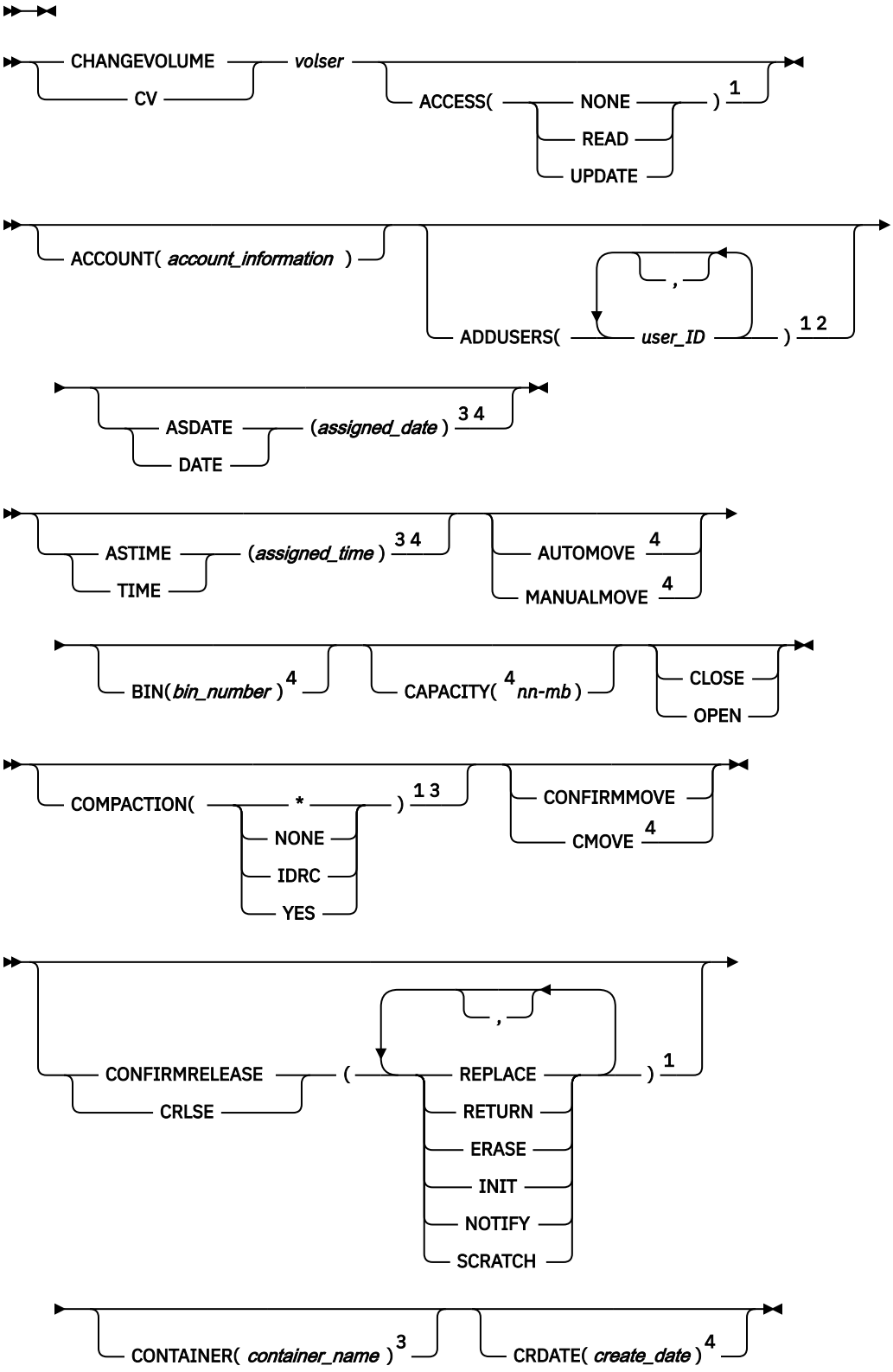
Confirmrelease parameters

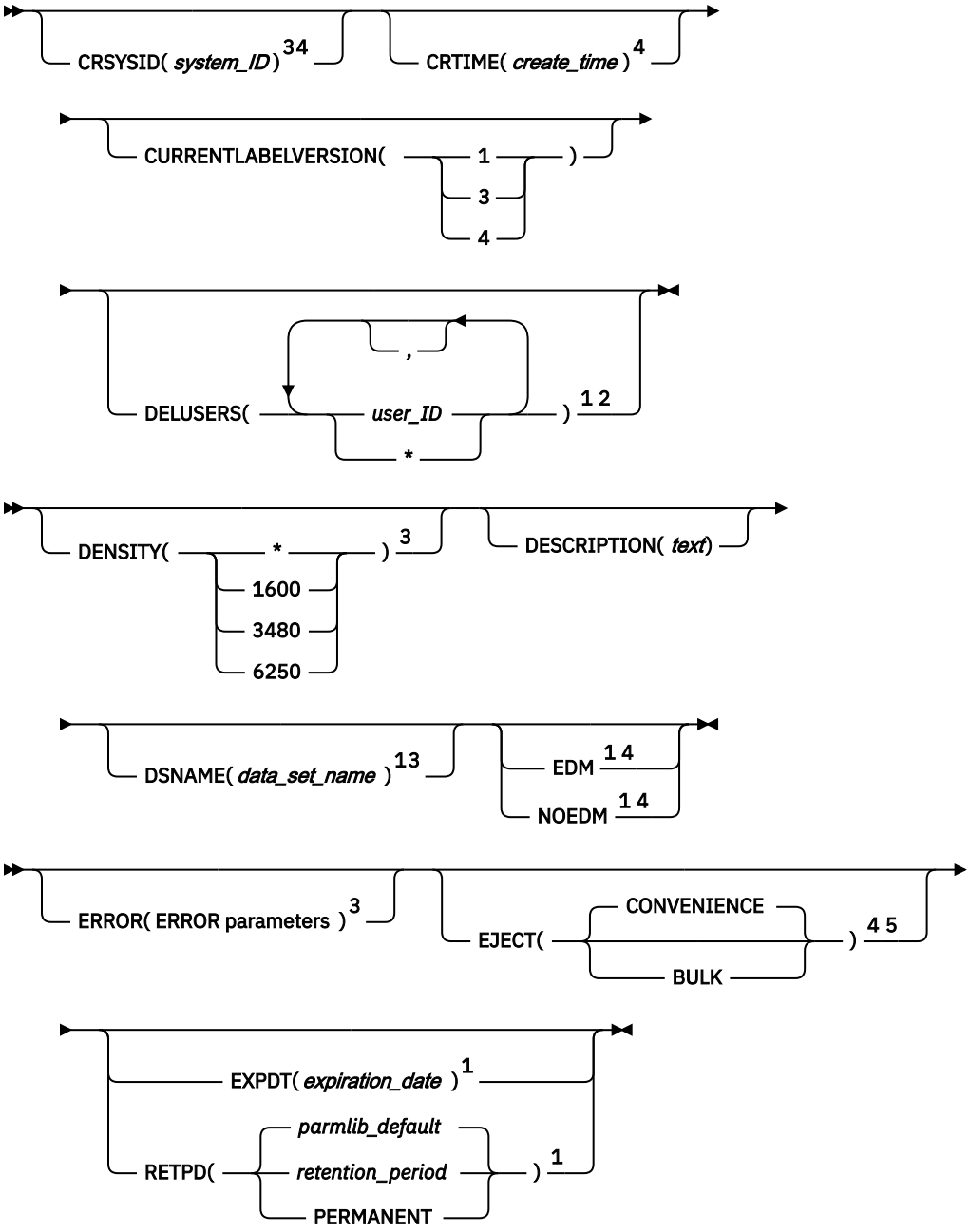


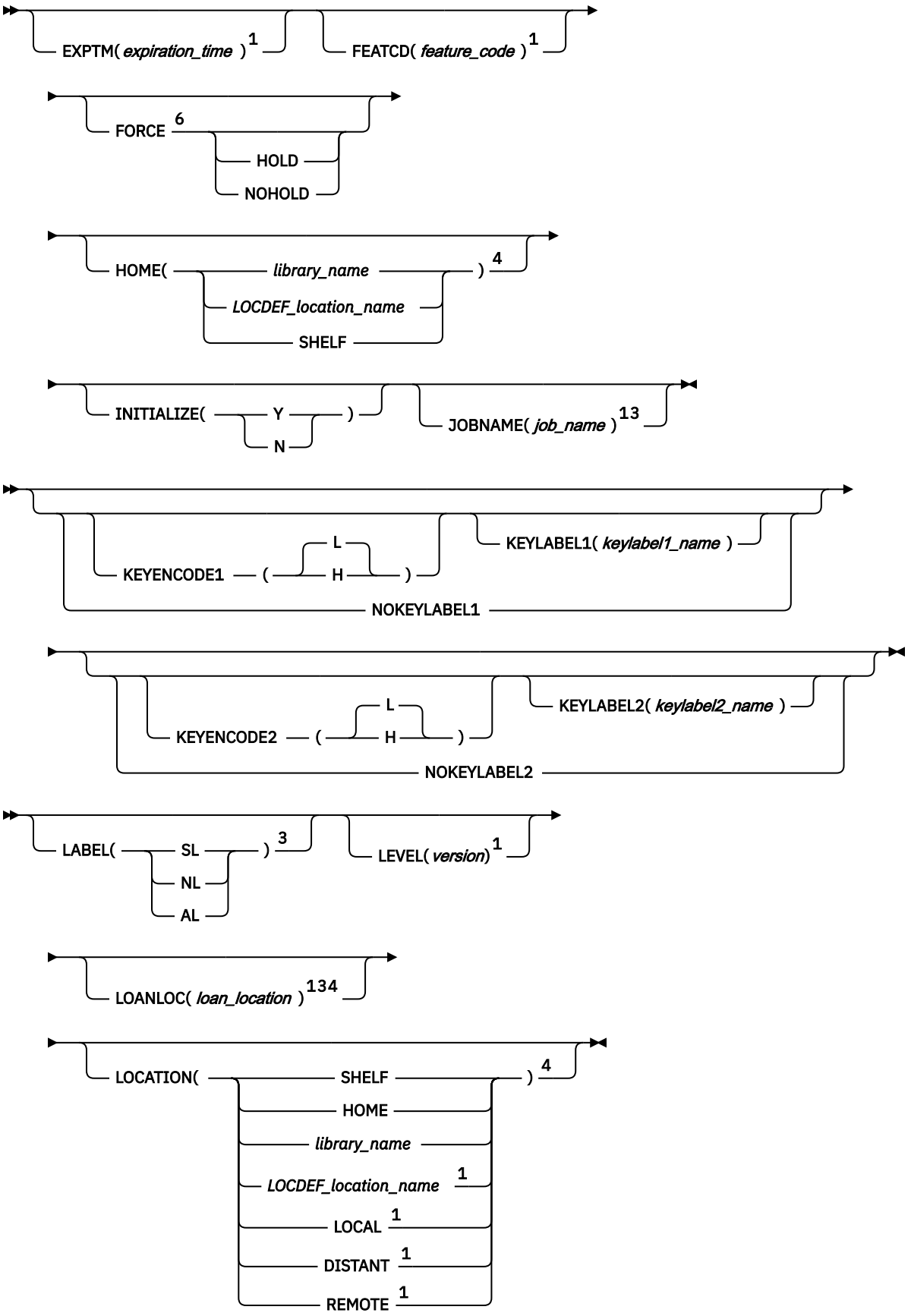
MOVE parameters

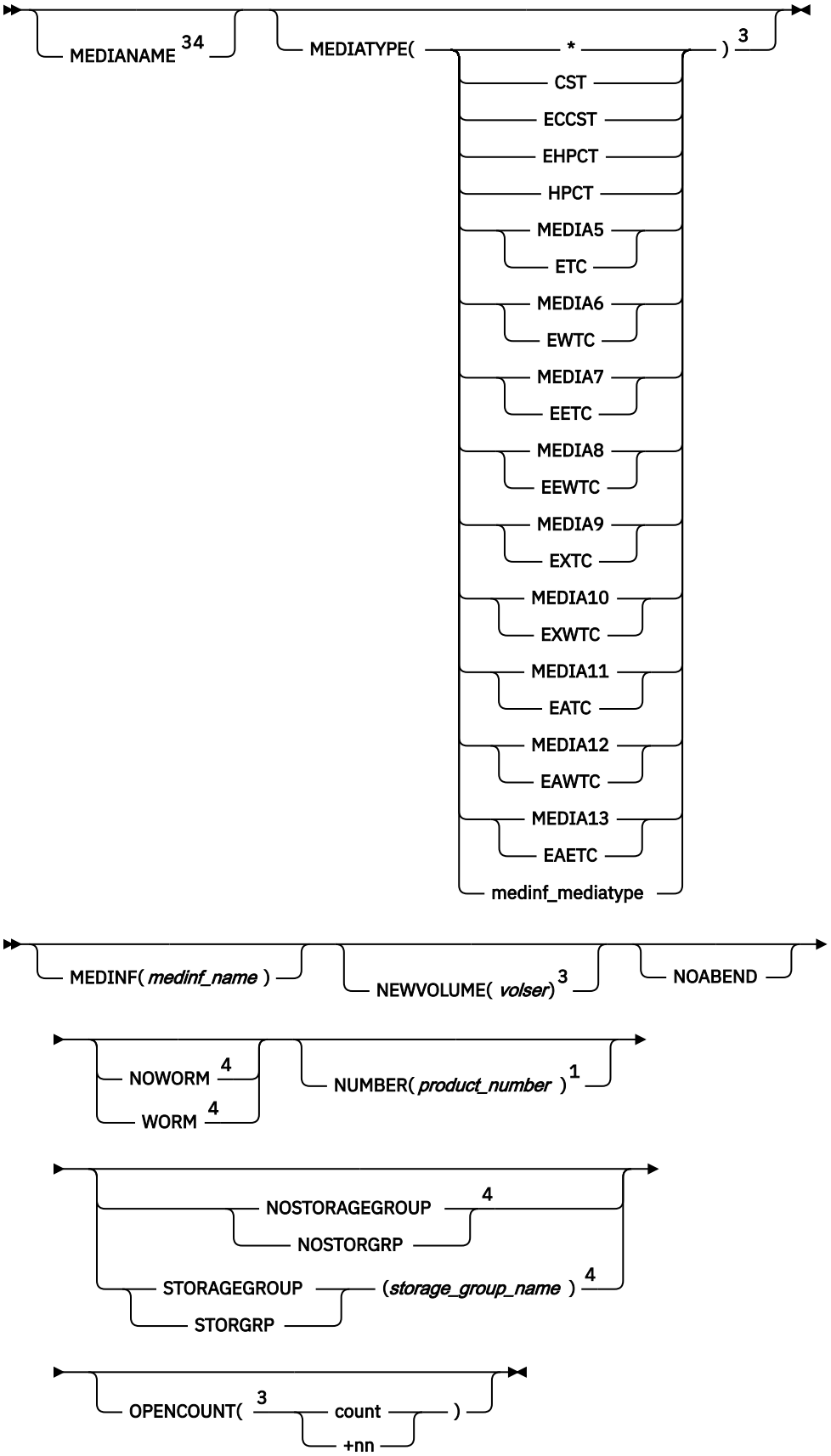


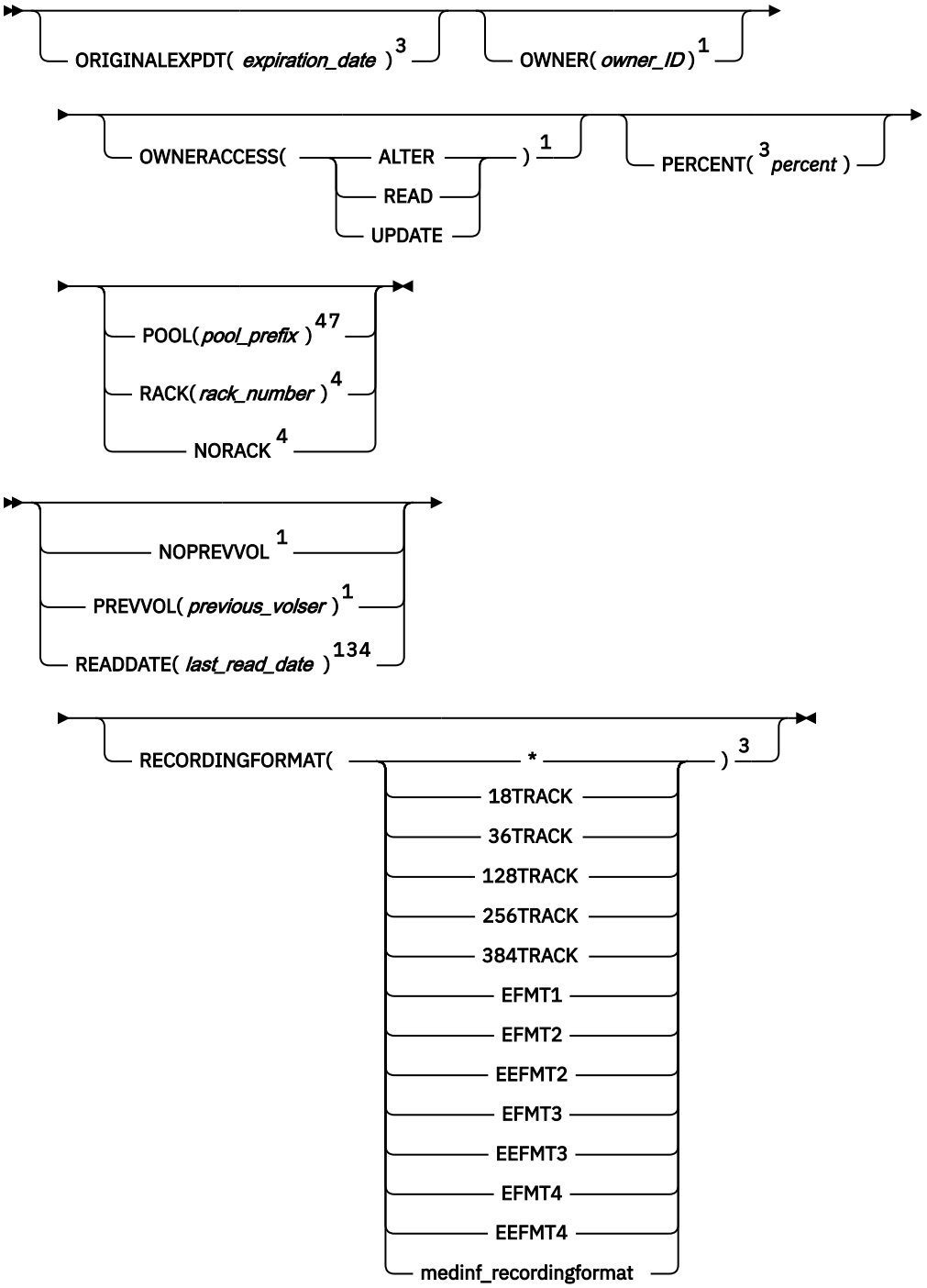
CHANGEVOLUME: Specific volume optional operands

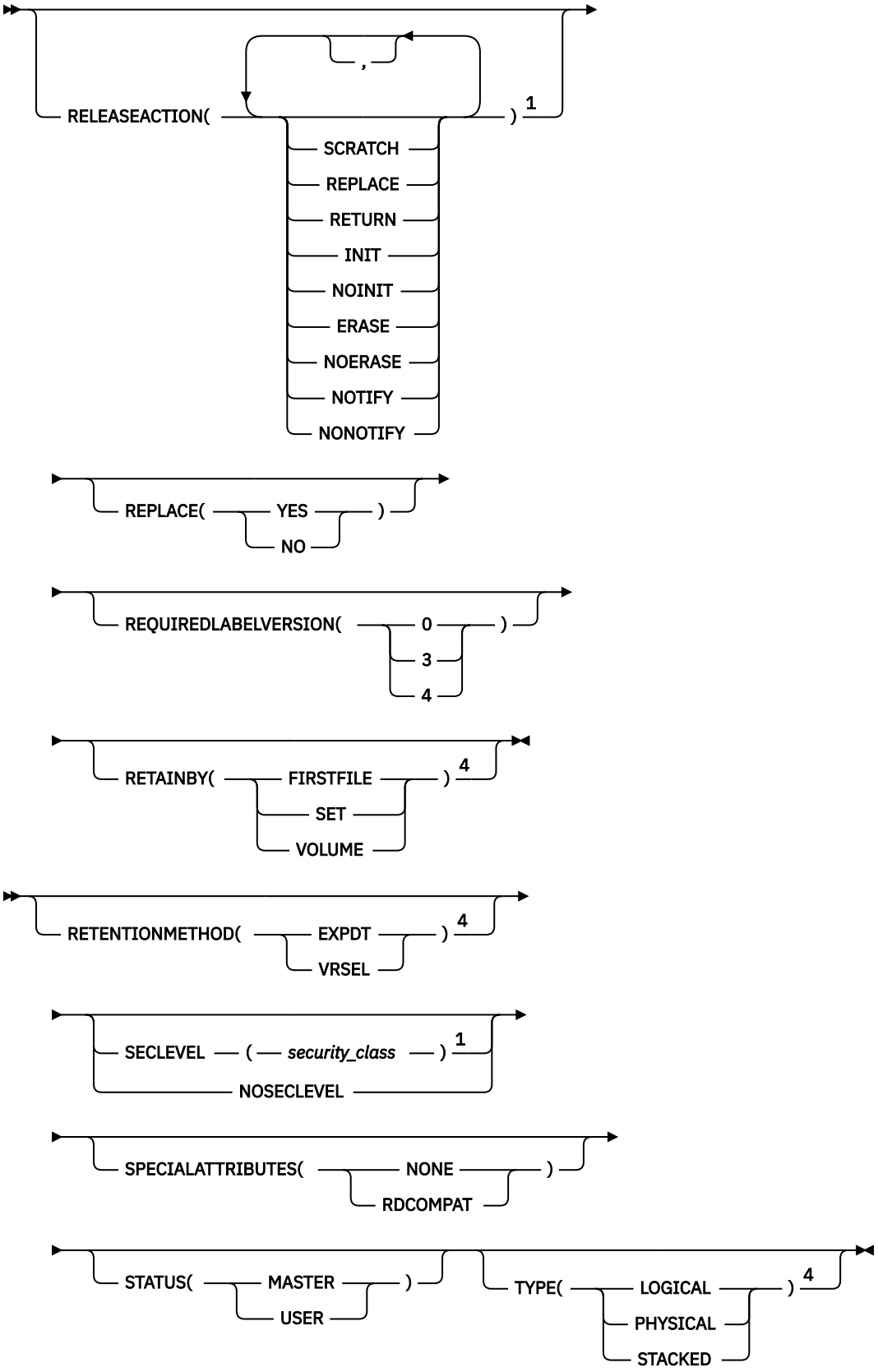


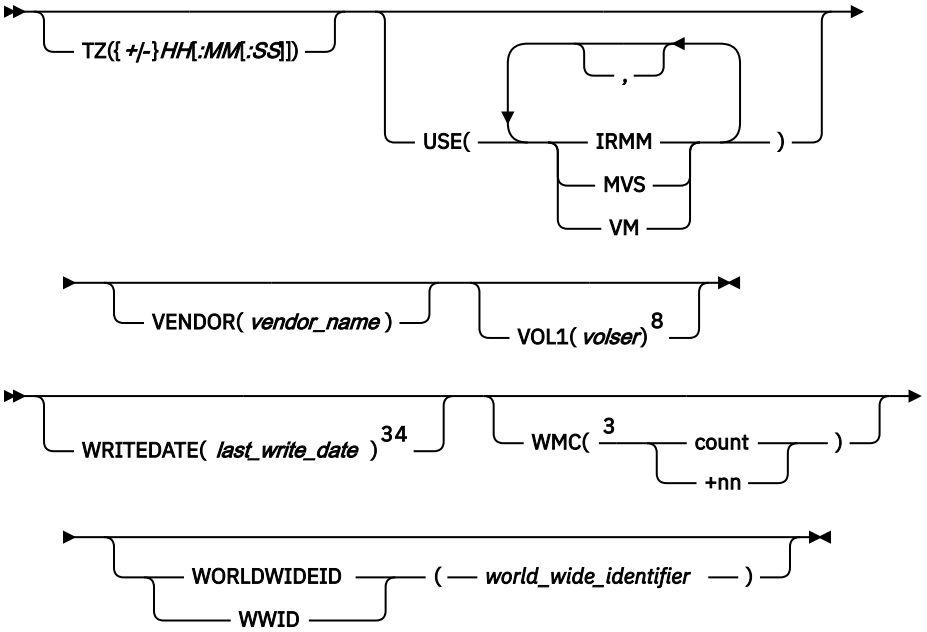

CRSYSID(system_ID) 3 4
CRTIME(create_time) 4
CURRENTLABELVERSION(1 3 4)
DELUSERS(, user_ID *) 1 2
DENSITY(* 1600 3480 6250) 3
DESCRIPTION(text)
DSNAME(data_set_name) 1 3
EDM 1 4
NOEDM 1 4
ERROR(ERROR parameters) 3
EJECT(CONVENIENCE BULK) 4 5
EXPDT(expiration_date) 1
RETPD(parmlib_default retention_period PERMANENT) 1


EXPTM(expiration_time) 1
FEATCD(feature_code) 1
FORCE 6
HOLD
NOHOLD
HOME(library_name) 4
LOCDEF_location_name
SHELF
INITIALIZE(Y) N
JOBNAME(job_name) 13
KEYENCODE1 (L H)
KEYLABEL1(keylabel1_name)
NOKEYLABEL1
KEYENCODE2 (L H)
KEYLABEL2(keylabel2_name)
NOKEYLABEL2
LABEL(SL) 3
NL
AL
LEVEL(version) 1
LOANLOC(loan_location) 134
LOCATION(SHELF) 4
HOME
library_name
LOCDEF_location_name 1
LOCAL 1
DISTANT 1
REMOTE 1

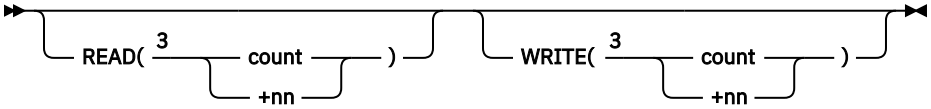

MEDIANAME 34
MEDIATYPE(*) 3
CST
ECCST
EHPCT
HPCT
MEDIA5
ETC
MEDIA6
EWTC
MEDIA7
EETC
MEDIA8
EEWTC
MEDIA9
EXTC
MEDIA10
EXWTC
MEDIA11
EATC
MEDIA12
EAWTC
MEDIA13
EAETC
medinf_mediatype
MEDINF(medinf_name)
NEWVOLUME(volser) 3
NOABEND
NOWORM 4
WORM 4
NUMBER(product_number) 1
NOSTORAGEGROUP 4
NOSTORGRP
STORAGEGROUP (storage_group_name) 4
STORGRP
OPENCOUNT(3 count)
+nn


ORIGINALEXPDT(expiration_date) 3
OWNER(owner_ID) 1
OWNERACCESS(ALTER READ UPDATE) 1
PERCENT(3 percent)
POOL(pool_prefix) 47
RACK(rack_number) 4
NORACK 4
NOPREVVOL 1
PREVVOL(previous_volser) 1
READDATE(last_read_date) 134
RECORDINGFORMAT(*) 3
18TRACK
36TRACK
128TRACK
256TRACK
384TRACK
EFMT1
EFMT2
EEFMT2
EFMT3
EEFMT3
EFMT4
EEFMT4
medinf_recordingformat


RELEASEACTION(
,
)
1
SCRATCH
REPLACE
RETURN
INIT
NOINIT
ERASE
NOERASE
NOTIFY
NONOTIFY
REPLACE(
YES
NO
)
REQUIREDLABELVERSION(
0
3
4
)
RETAINBY(
FIRSTFILE
SET
VOLUME
)
4
RETENTIONMETHOD(
EXPDT
VRSEL
)
4
SECLEVEL
(
security_class
)
1
NOSECLEVEL
SPECIALATTRIBUTES(
NONE
RDCOMPAT
)
STATUS(
MASTER
USER
)
TYPE(
LOGICAL
PHYSICAL
STACKED
)
4



ERROR parameters



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.

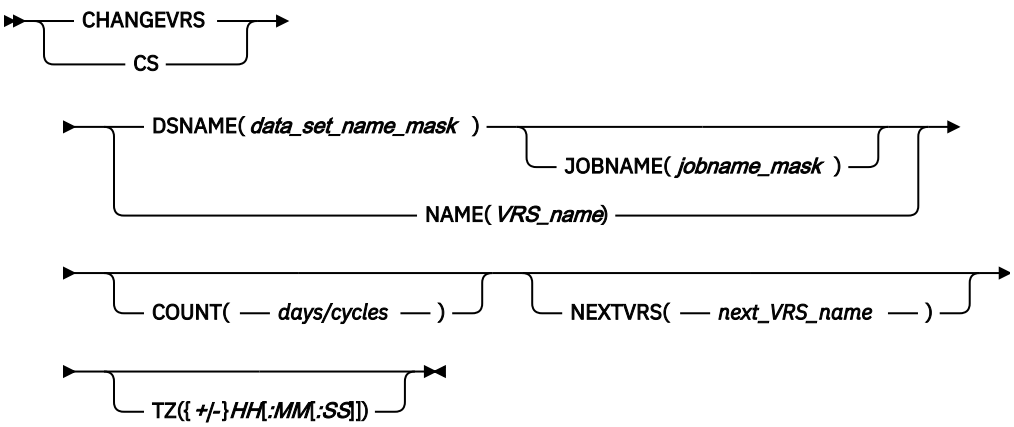
[5] This operand can be specified only if the volume resides in a system-managed library.

[6] This operand can be specified to change a data value 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.

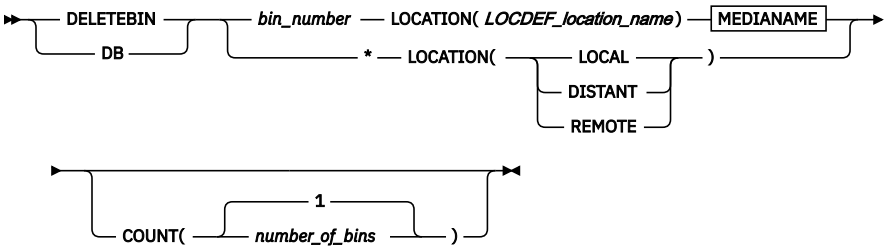
[7] This operand cannot be specified if the volume resides in an IBM Tape Library Dataserver.

[8] This operand cannot be specified for SCRATCH volumes, LOGICAL volumes, STACKED volumes, or NOLABEL volumes.

CHANGEVRS: Changing a vital record specification

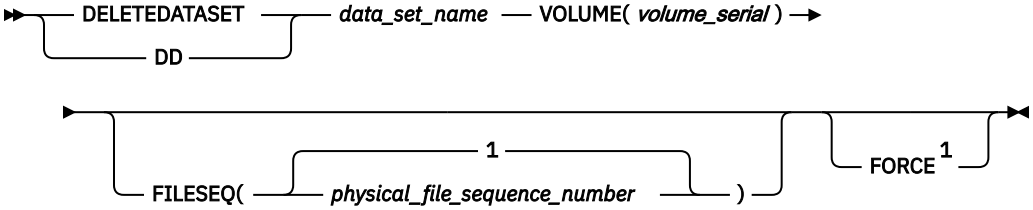


DELETEBIN: Deleting bin number information



MEDIANAME

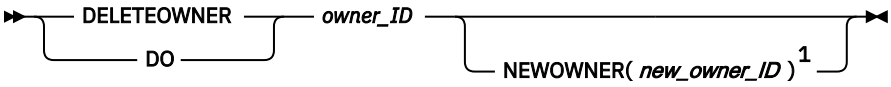
DELETEDATASET: Deleting data set information



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.

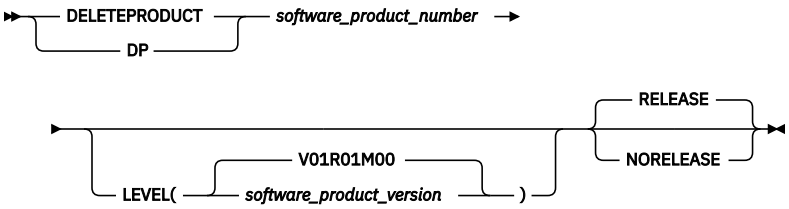
DELETEOWNER: Deleting owner information



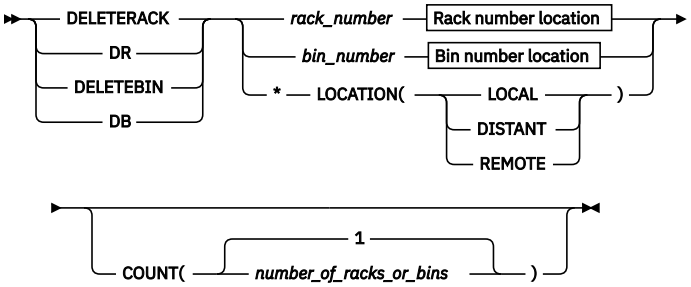
Notes:

[1] The NEWOWNER operand specified to reassign volumes if the owner you are deleting owns one or more volumes.

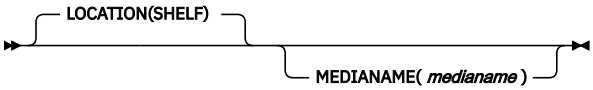
DELETEPRODUCT: Deleting software product information



DELETERACK: Deleting shelf location information

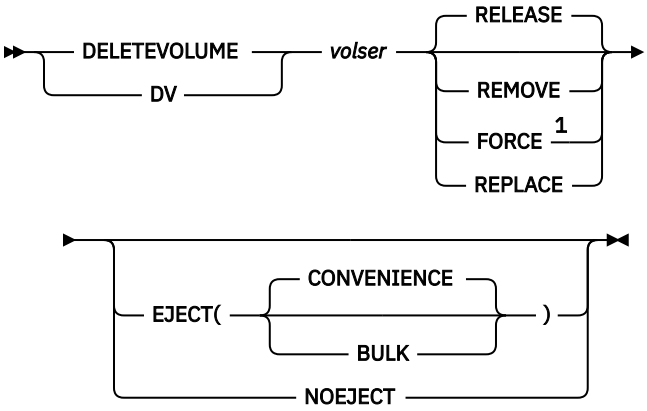


Rack number location



Bin number location

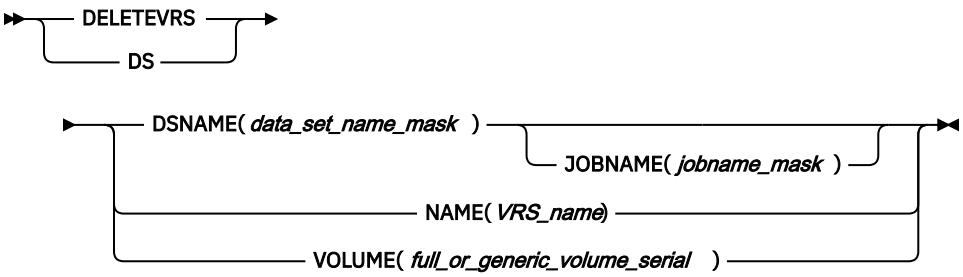
DELETEVOLUME: Deleting volume information



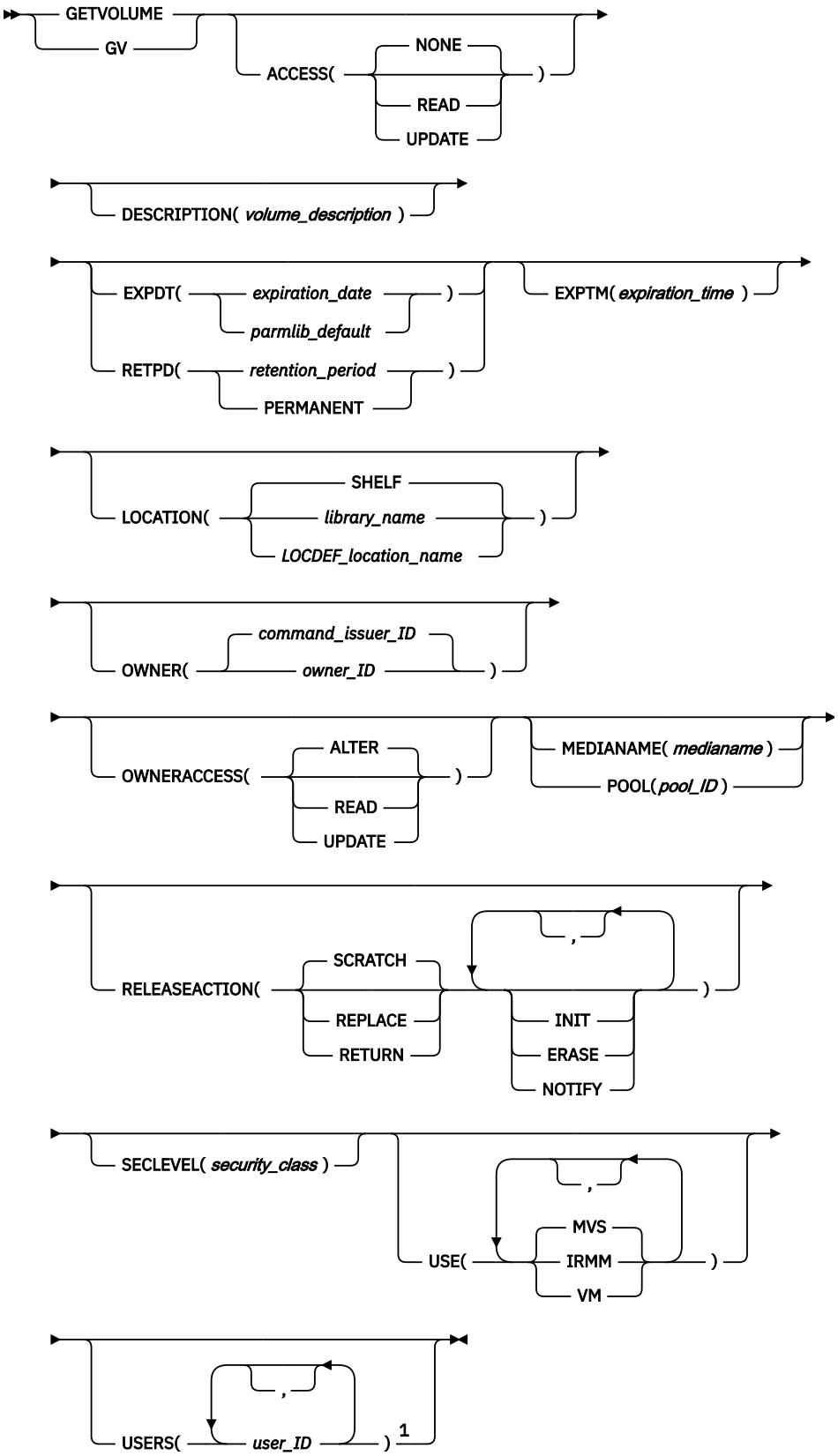
Notes:

[1] You must have CONTROL access to STGADMIN.EDG.MASTER security resource and UPDATE access to STGADMIN.EDG.FORCE security resource to use the FORCE operand.

DELETEVRS: Deleting vital record specification information



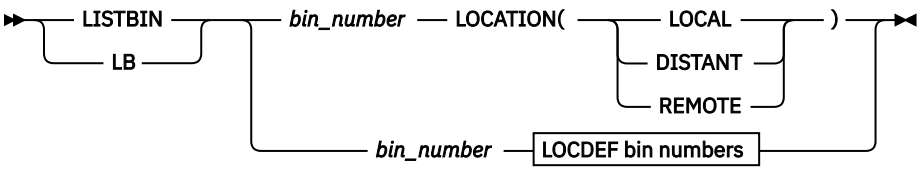
GETVOLUME: Requesting and assigning scratch volumes



Notes:

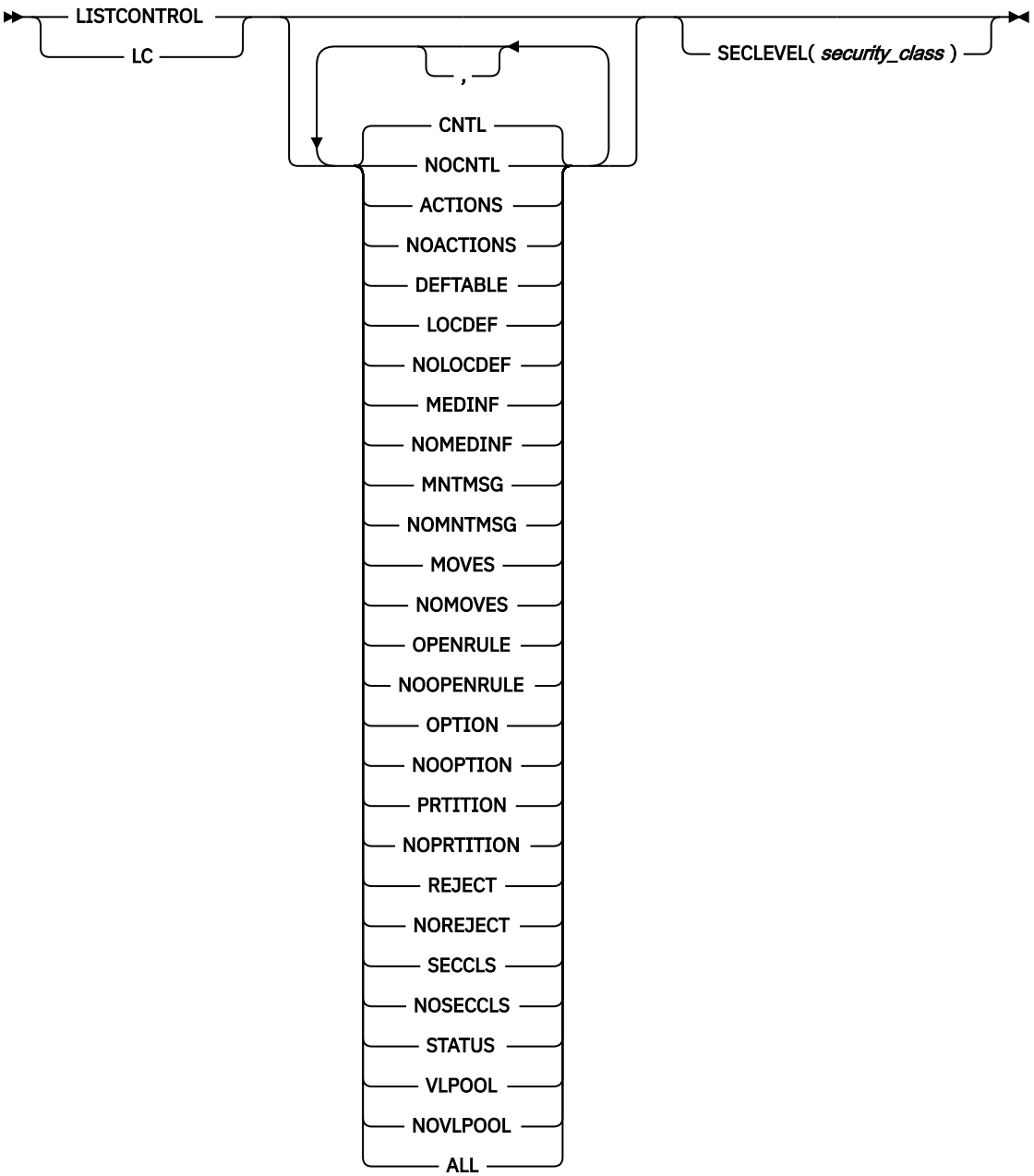
[1] You can specify a maximum of 12 user IDs.

LISTBIN: Displaying information about a shelf location

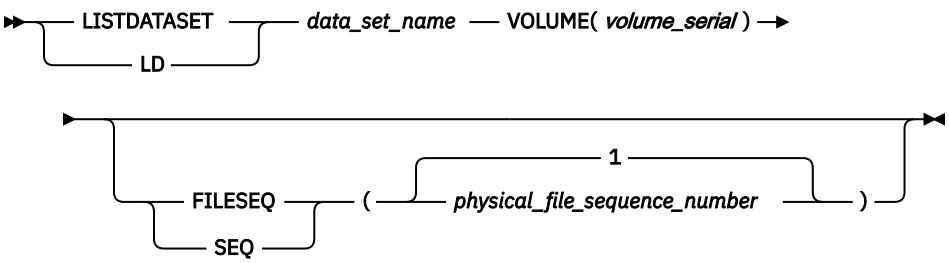


LOCDEF bin numbers

LISTCONTROL: Displaying parmlib options and control information

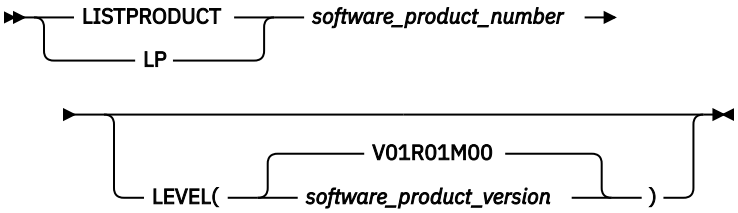


LISTDATASET: Displaying information about a data set

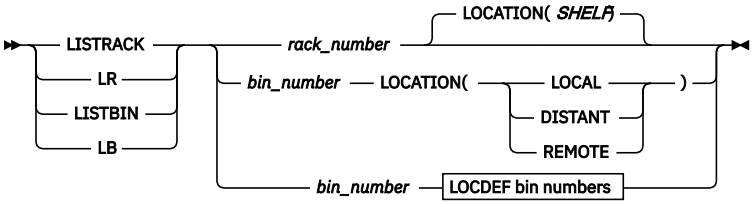


LISTOWNER: Displaying information about an owner

LISTPRODUCT: Displaying information about a software product

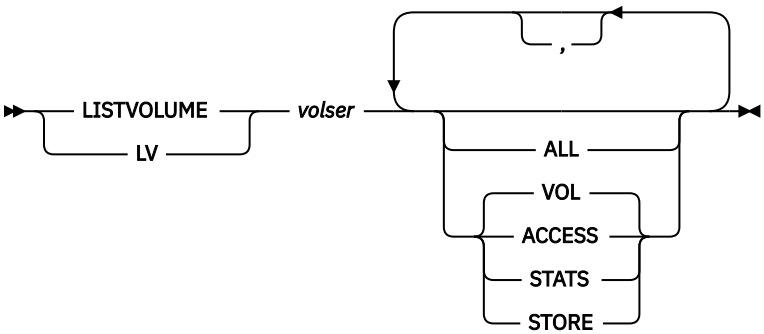


LISTRACK: Displaying information about a shelf location

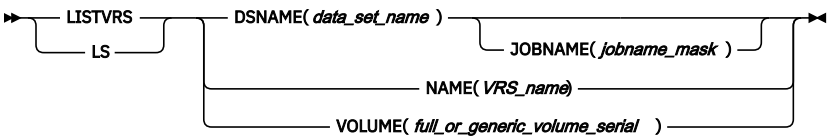


LOCDEF bin numbers

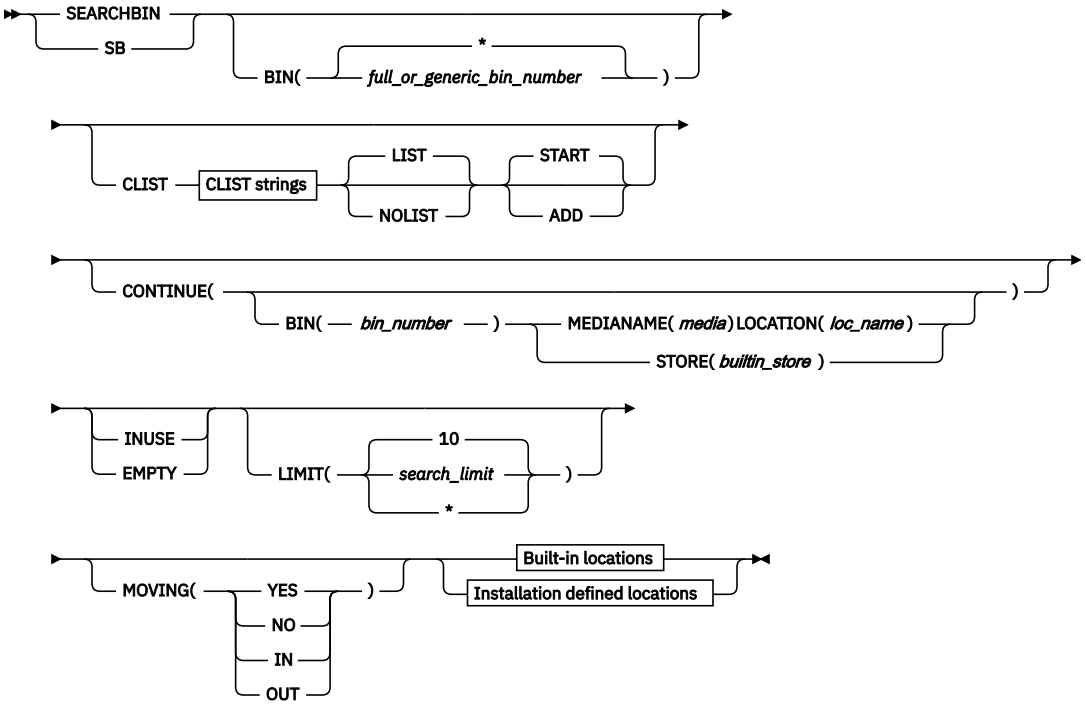
LISTVOLUME: Displaying information about a volume



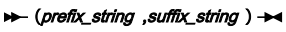
LISTVRS: Displaying information about a vital record specification



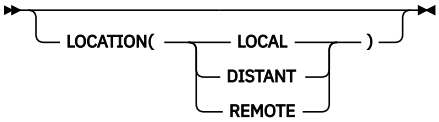
SEARCHBIN: Creating a list of bin numbers



CLIST strings

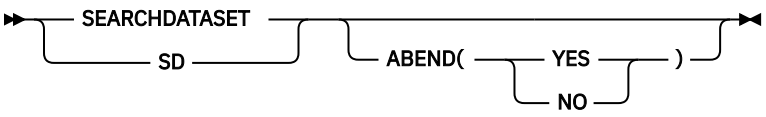


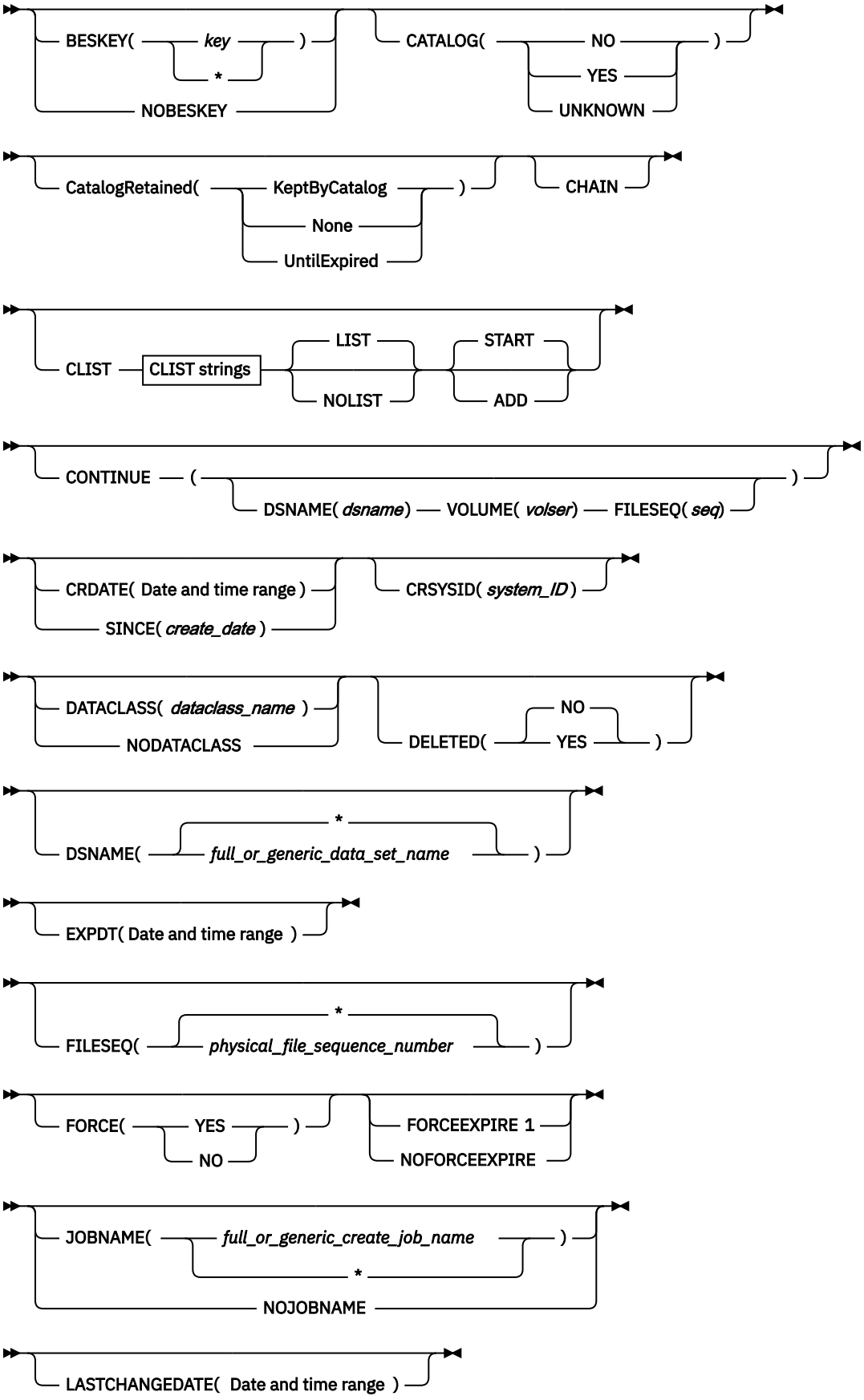
Built-in locations

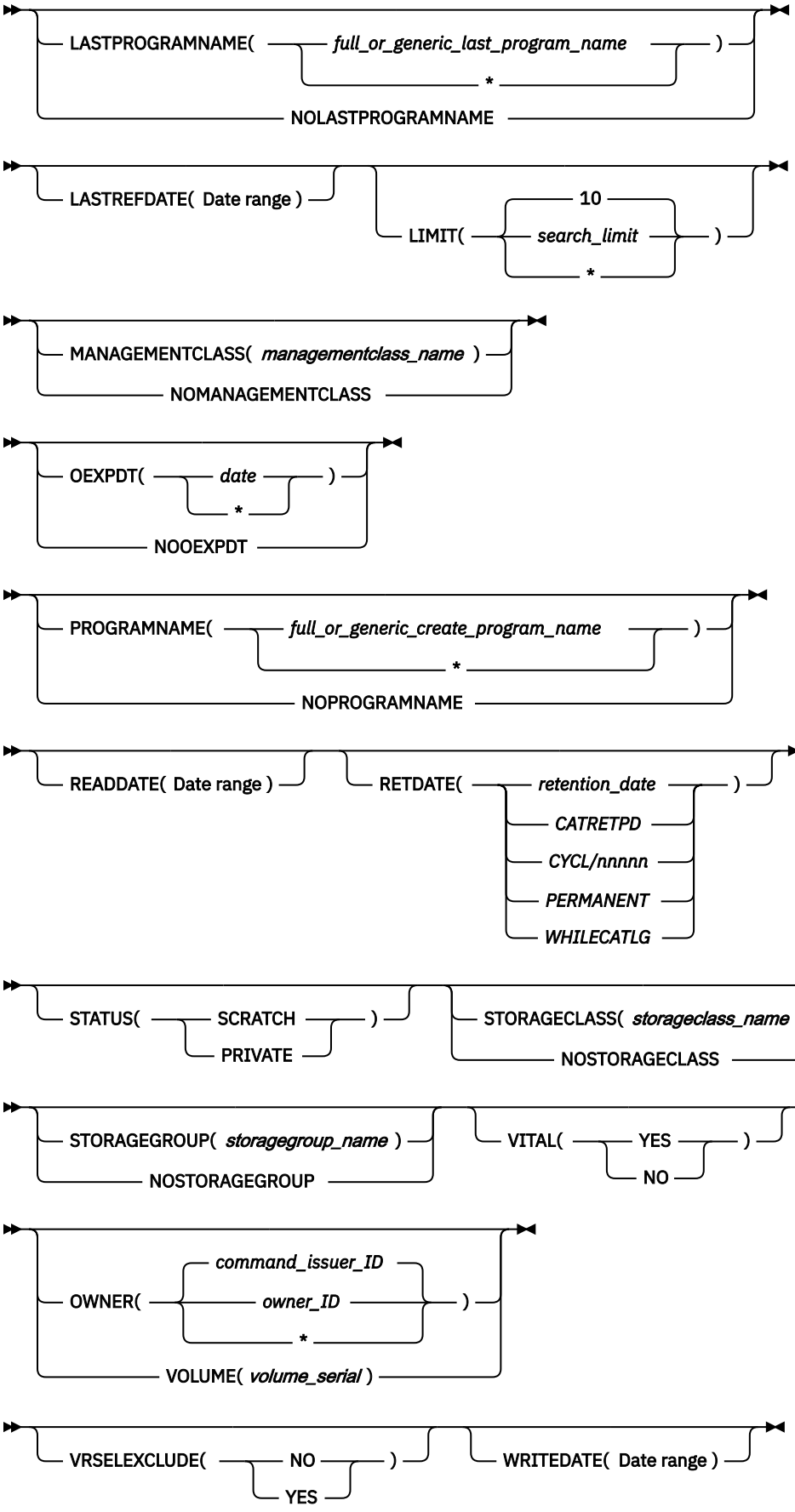


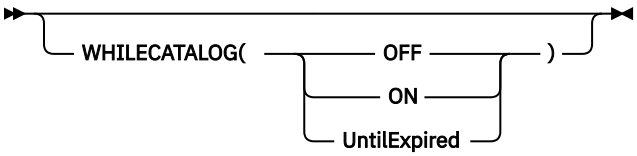
Installation defined locations

SEARCHDATASET: Creating a list of data sets

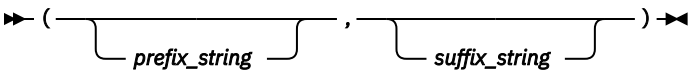



BESKEY(key *) NOBESKEY
CATALOG(NO YES UNKNOWN)
CatalogRetained(KeptByCatalog None UntilExpired)
CHAIN
CLIST CLIST strings LIST NOLIST START ADD
CONTINUE (DSNAME(dsname) VOLUME(volser) FILESEQ(seq))
CRDATE(Date and time range) SINCE(create_date)
CRSYSID(system_ID)
DATACLASS(dataclass_name) NODATACLASS
DELETED(NO YES)
DSNAME(* full_or_generic_data_set_name)
EXPDT(Date and time range)
FILESEQ(* physical_file_sequence_number)
FORCE(YES NO)
FORCEEXPIRE 1 NOFORCEEXPIRE
JOBNAME(full_or_generic_create_job_name *) NOJOBNAME
LASTCHANGEDATE(Date and time range)

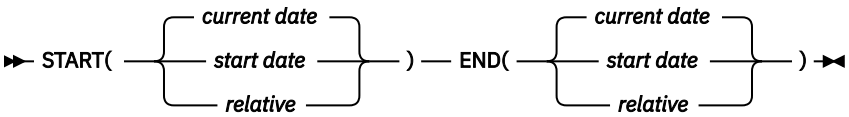

LASTPROGRAMNAME(full_or_generic_last_program_name)
*
NOLASTPROGRAMNAME
LASTREFDATE(Date range)
LIMIT(10 search_limit *)
MANAGEMENTCLASS(managementclass_name)
NOMANAGEMENTCLASS
OEXPDT(date)
*
NOOEXPDT
PROGRAMNAME(full_or_generic_create_program_name)
*
NOPROGRAMNAME
READDATE(Date range)
RETDATE(retention_date CATRETPD CYCL/nnnnn PERMANENT WHILECATLG)
STATUS(SCRATCH PRIVATE)
STORAGECLASS(storageclass_name)
NOSTORAGECLASS
STORAGEGROUP(storagegroup_name)
NOSTORAGEGROUP
VITAL(YES NO)
OWNER(command_issuer_ID owner_ID *)
VOLUME(volume_serial)
VRSELEXCLUDE(NO YES)
WRITEDATE(Date range)



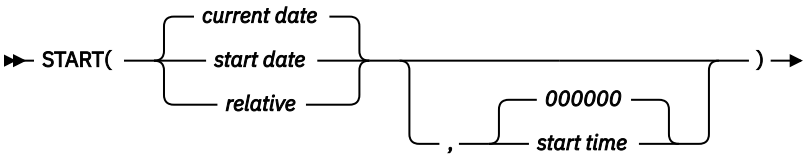
CLIST strings

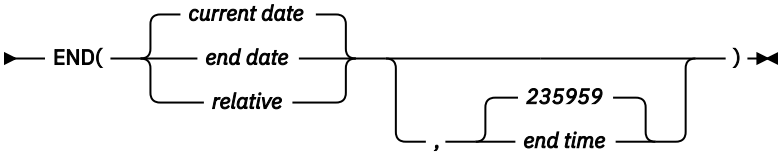


Date range

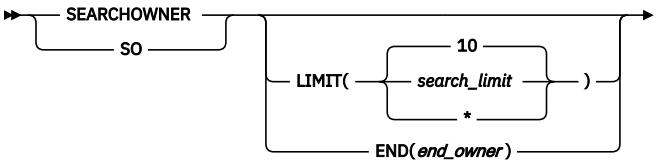


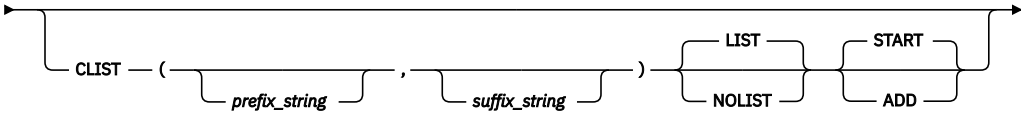
Date and time range

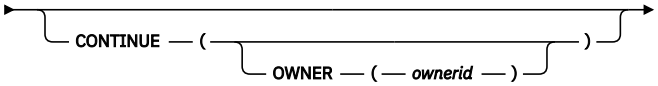


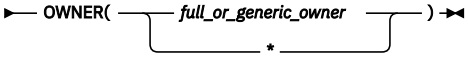


SEARCHOWNER: Creating a list of owner information

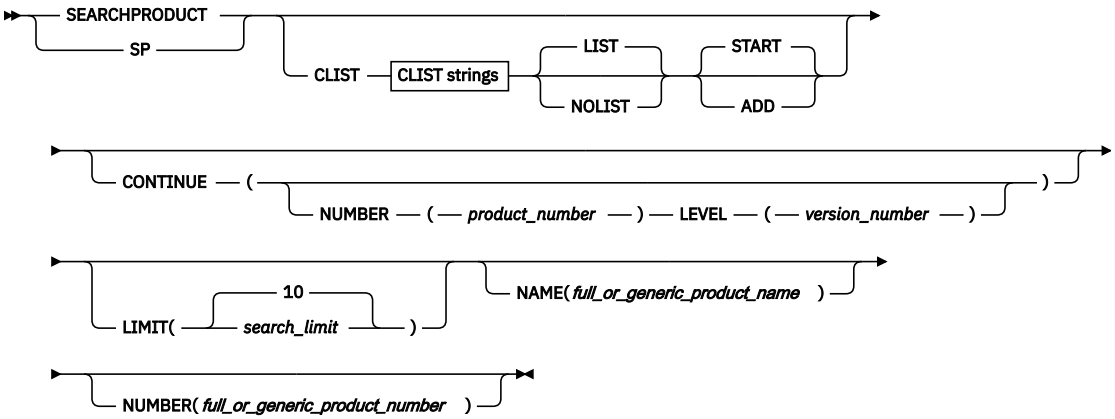






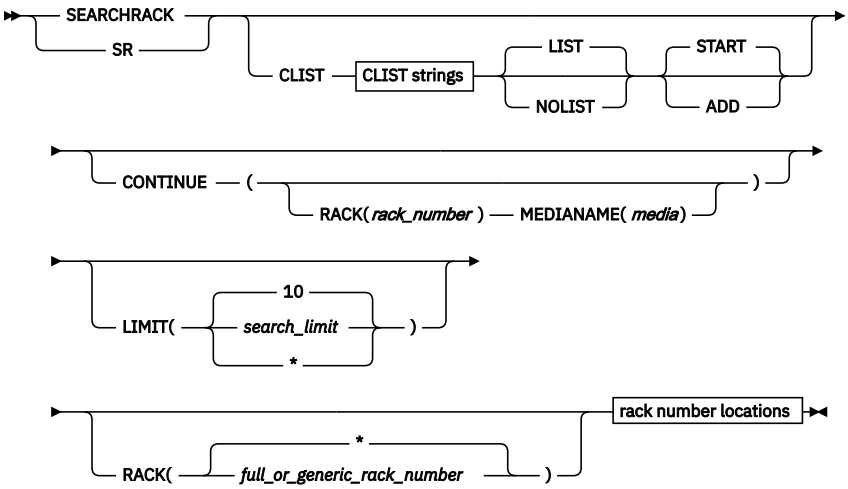


SEARCHPRODUCT: Creating a list of software products



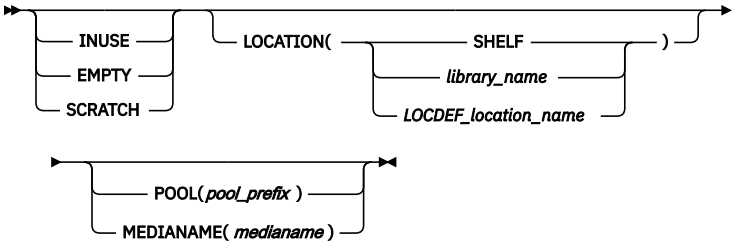
CLIST strings

SEARCHRACK: Creating a list of shelf locations

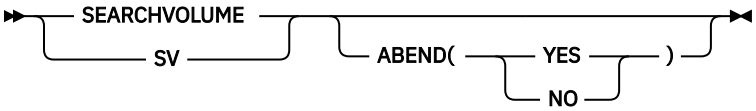


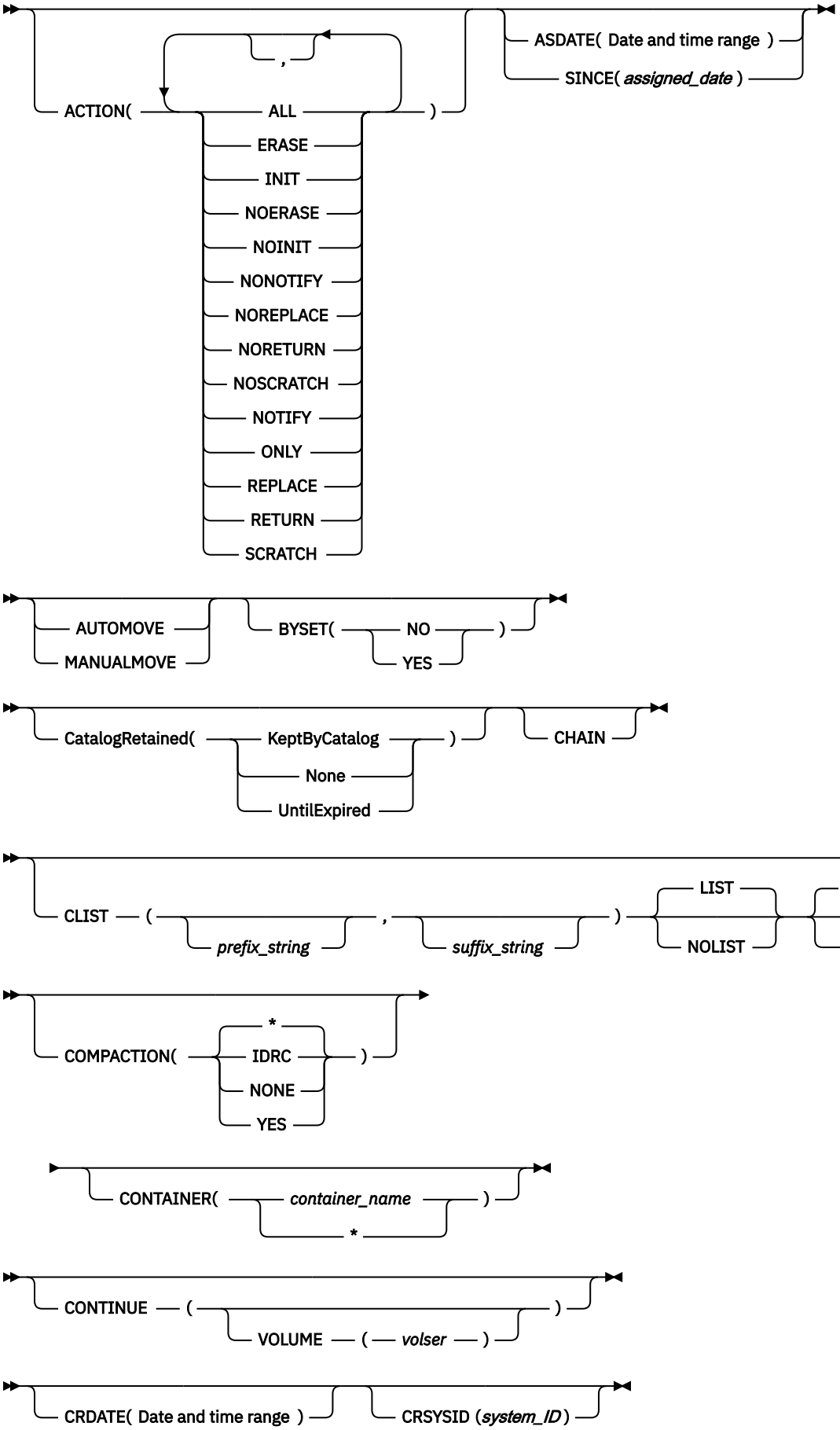
CLIST strings

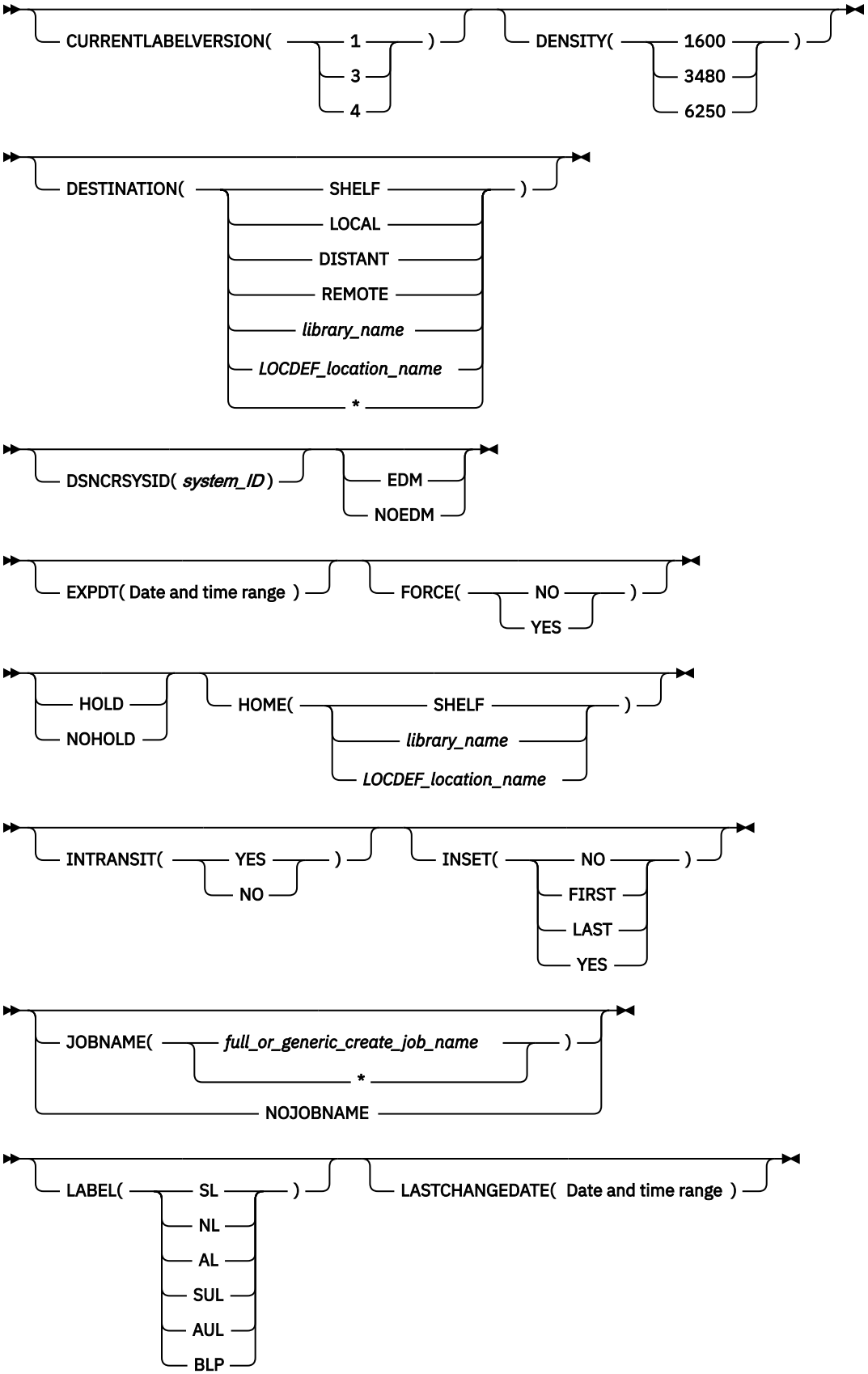
rack number locations

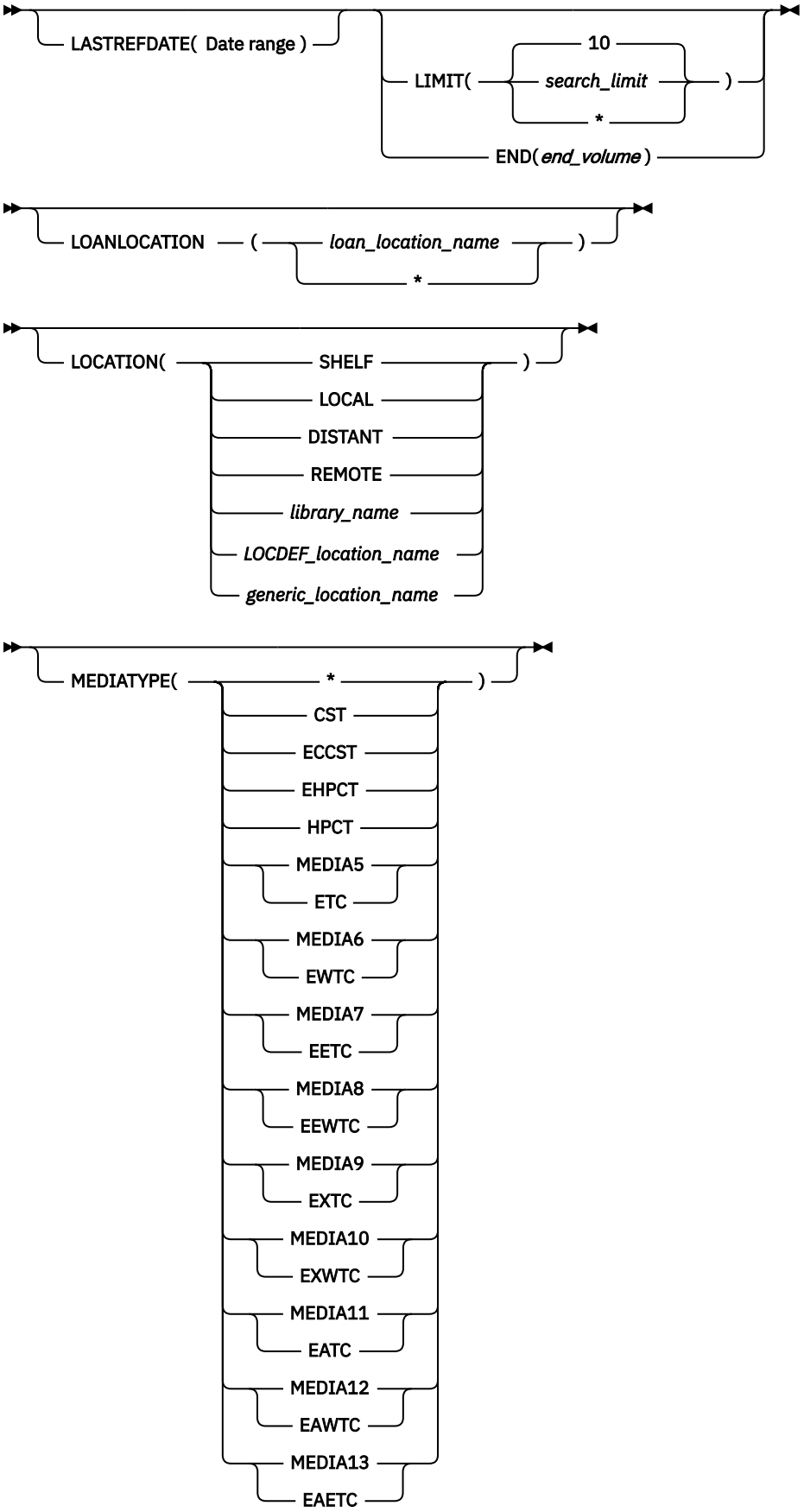


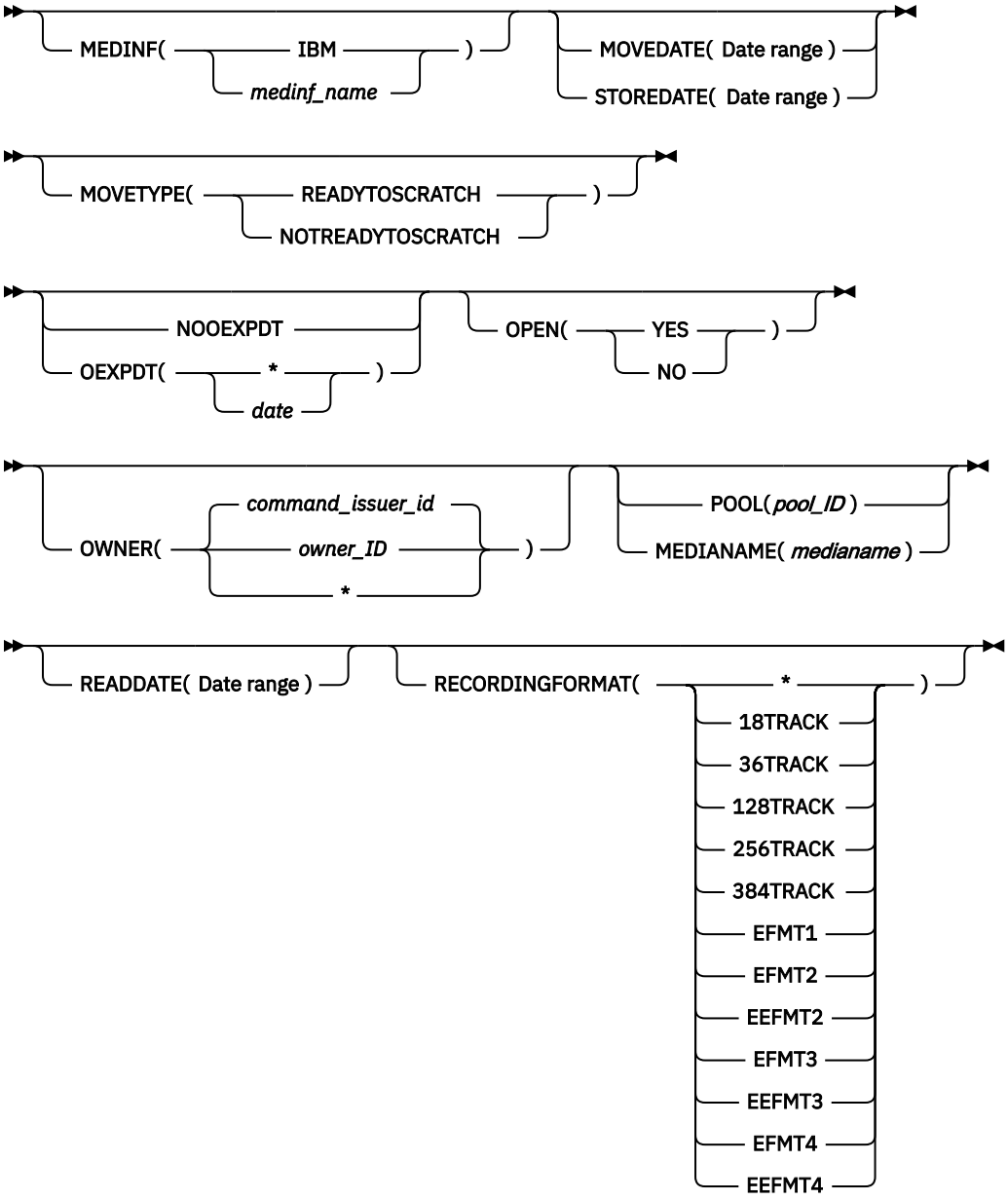
SEARCHVOLUME: Creating a list of volumes

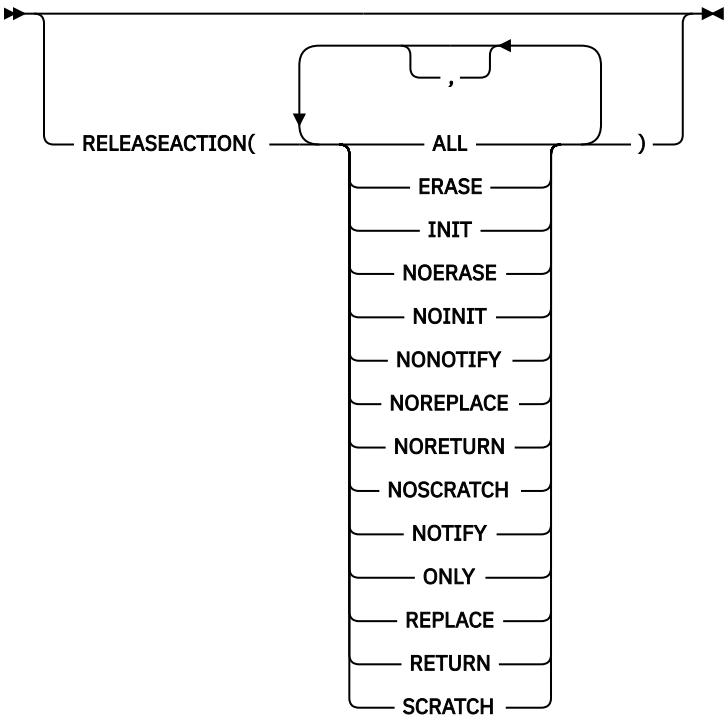


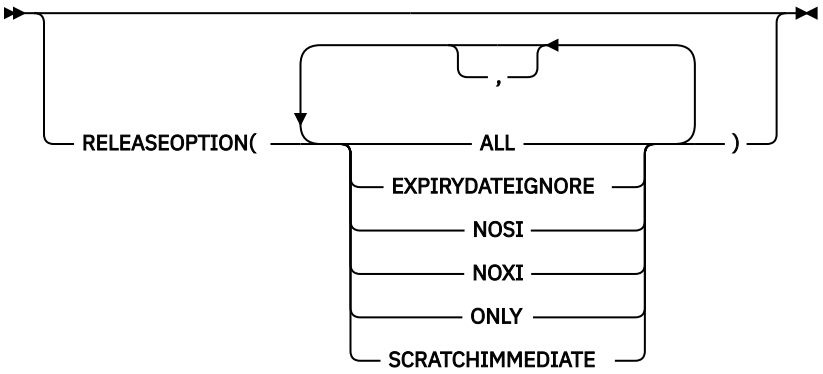

ACTION(
,
ALL
ERASE
INIT
NOERASE
NOINIT
NONOTIFY
NOREPLACE
NORETURN
NOSCRATCH
NOTIFY
ONLY
REPLACE
RETURN
SCRATCH
)
ASDATE(Date and time range)
SINCE(assigned_date)
AUTOMOVE
MANUALMOVE
BYSET(
NO
YES
)
CatalogRetained(
KeptByCatalog
None
UntilExpired
)
CHAIN
CLIST
(
prefix_string
,
suffix_string
)
LIST
NOLIST
START
ADD
COMPACTION(
*
IDRC
NONE
YES
)
CONTAINER(
container_name
*
)
CONTINUE
(
VOLUME
(
volser
)
)
CRDATE(Date and time range)
CRSYSID (system_ID)

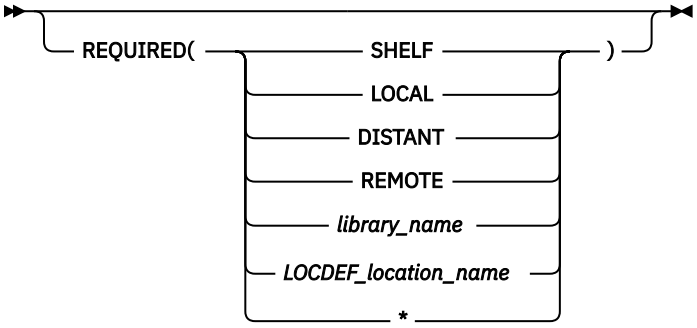

CURRENTLABELVERSION(1 3 4)
DENSITY(1600 3480 6250)
DESTINATION(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name *)
DSNCRSYSID(system_ID) EDM NOEDM
EXPDT(Date and time range)
FORCE(NO YES)
HOLD NOHOLD
HOME(SHELF library_name LOCDEF_location_name)
INTRANSIT(YES NO)
INSET(NO FIRST LAST YES)
JOBNAME(full_or_generic_create_job_name *)
NOJOBNAME
LABEL(SL NL AL SUL AUL BLP)
LASTCHANGEDATE(Date and time range)

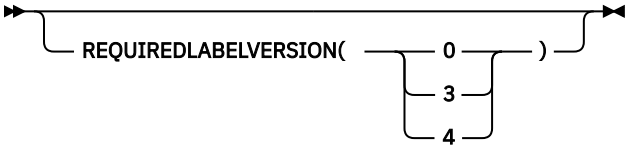

LASTREFDATE(Date range)
LIMIT(10 search_limit *)
END(end_volume)
LOANLOCATION (loan_location_name *)
LOCATION(SHELF LOCAL DISTANT REMOTE library_name LOCDEF_location_name generic_location_name)
MEDIATYPE(* CST ECCST EHPCT HPCT MEDIA5 ETC MEDIA6 EWTC MEDIA7 EETC MEDIA8 EEWTC MEDIA9 EXTC MEDIA10 EXWTC MEDIA11 EATC MEDIA12 EAWTC MEDIA13 EAETC)

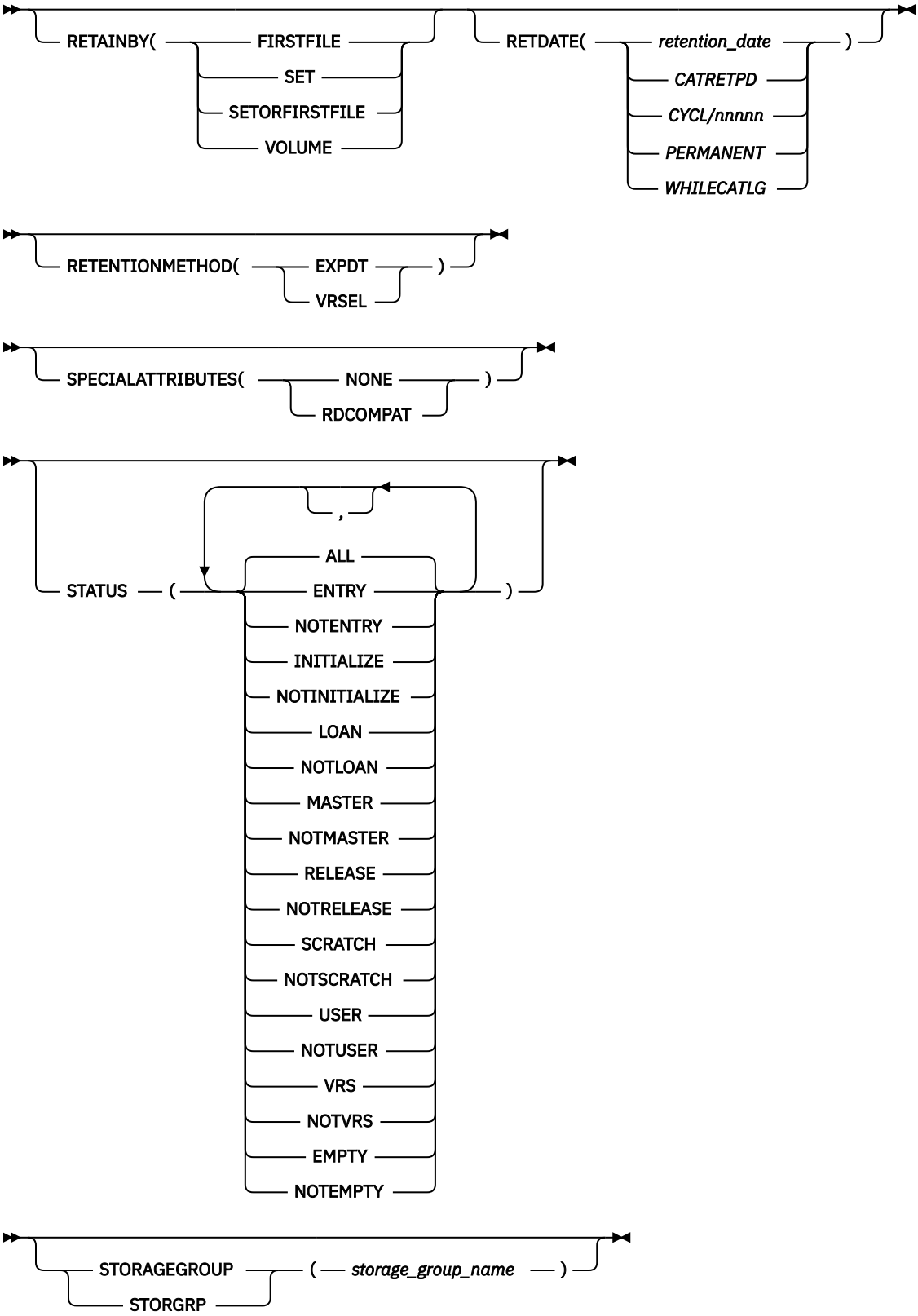

MEDINF(IBM) medinf_name
MOVEDATE(Date range)
STOREDATE(Date range)
MOVETYPE(READYTOSCRATCH) NOTREADYTOSCRATCH
NOOEXPDT
OEXPDT(*) date
OPEN(YES) NO
OWNER(command_issuer_id owner_ID *)
POOL(pool_ID)
MEDIANAME(medianame)
READDATE(Date range)
RECORDINGFORMAT(*)
18TRACK
36TRACK
128TRACK
256TRACK
384TRACK
EFMT1
EFMT2
EEFMT2
EFMT3
EEFMT3
EFMT4
EEFMT4

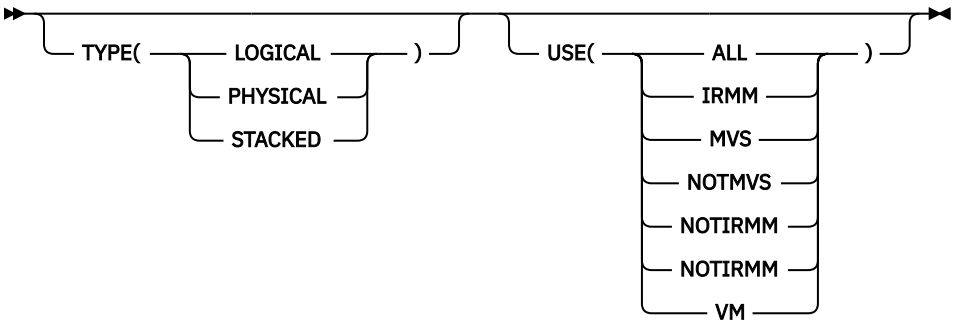

RELEASEACTION(
,
ALL
)
ERASE
INIT
NOERASE
NOINIT
NONOTIFY
NOREPLACE
NORETURN
NOSCRATCH
NOTIFY
ONLY
REPLACE
RETURN
SCRATCH

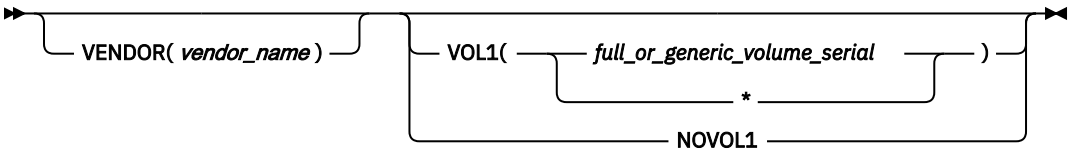

RELEASEOPTION(
,
ALL
)
EXPIRYDATEIGNORE
NOSI
NOXI
ONLY
SCRATCHIMMEDIATE

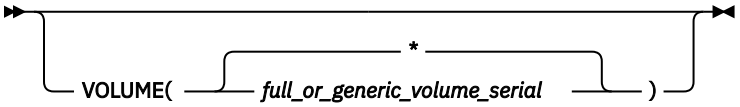

REQUIRED(
SHELF
)
LOCAL
DISTANT
REMOTE
library_name
LOCDEF_location_name
*

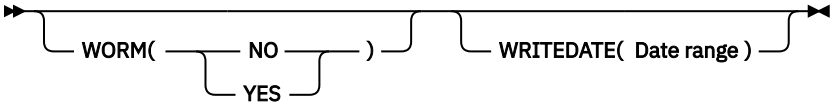

REQUIREDLABELVERSION(
0
)
3
4


RETAINBY(
FIRSTFILE
SET
SETORFIRSTFILE
VOLUME
RETDATE(
retention_date
CATRETPD
CYCL/nnnnn
PERMANENT
WHILECATLG
)
RETENTIONMETHOD(
EXPDT
VRSEL
)
SPECIALATTRIBUTES(
NONE
RDCOMPAT
)
STATUS
(
,
ALL
ENTRY
NOTENTRY
INITIALIZE
NOTINITIALIZE
LOAN
NOTLOAN
MASTER
NOTMASTER
RELEASE
NOTRELEASE
SCRATCH
NOTSCRATCH
USER
NOTUSER
VRS
NOTVRS
EMPTY
NOTEMPTY
)
STORAGEGROUP
STORGRP
(
storage_group_name
)

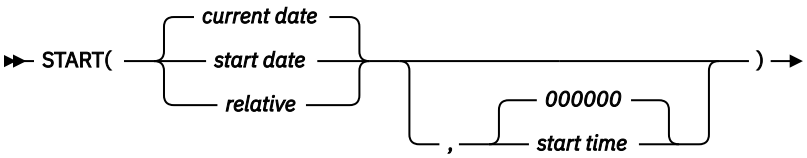


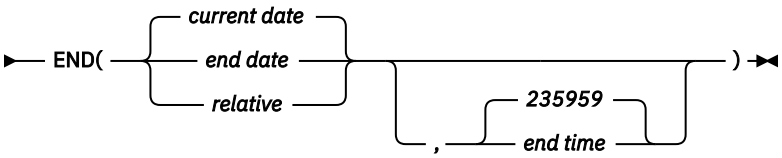




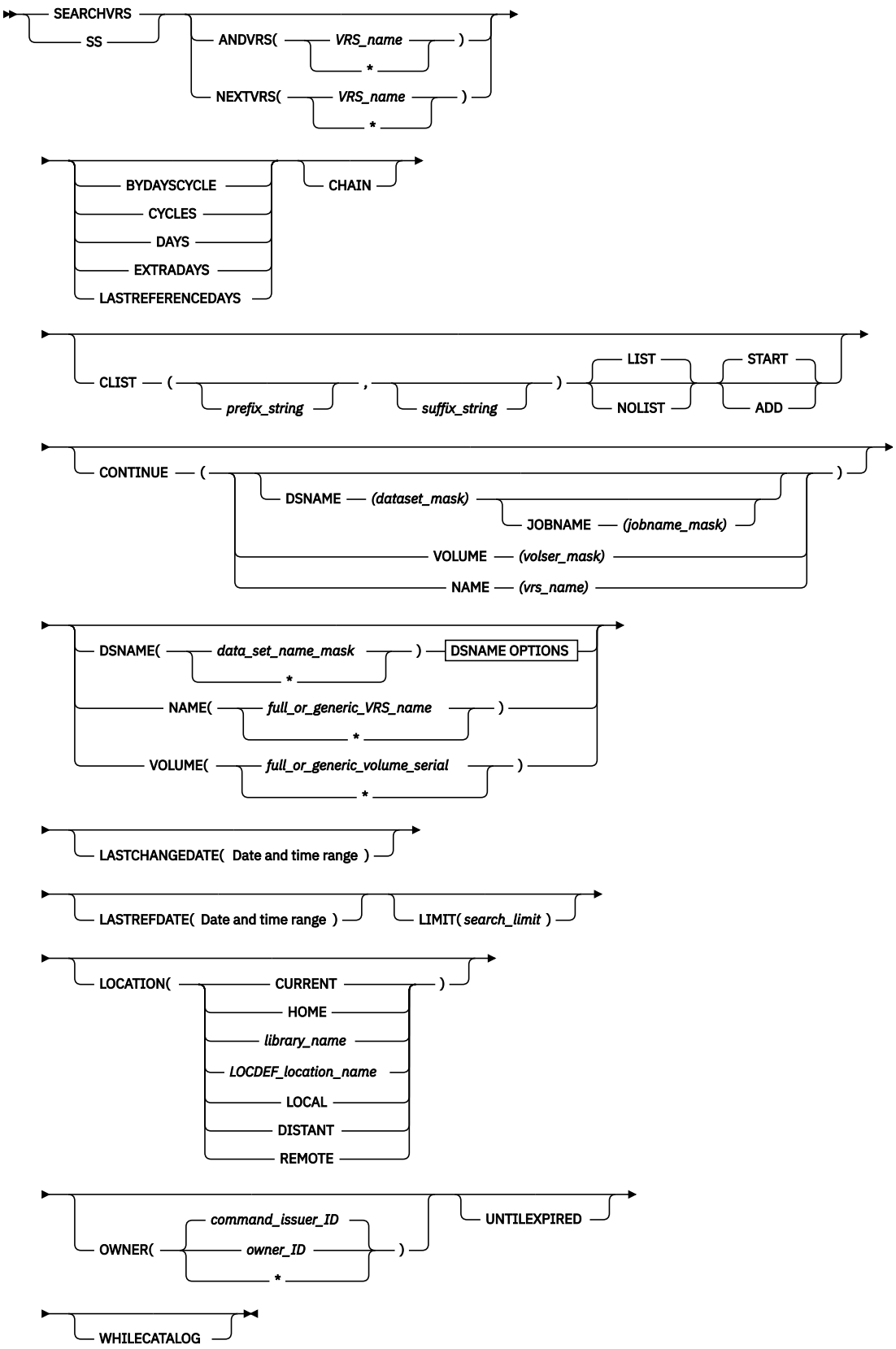


Date and time range

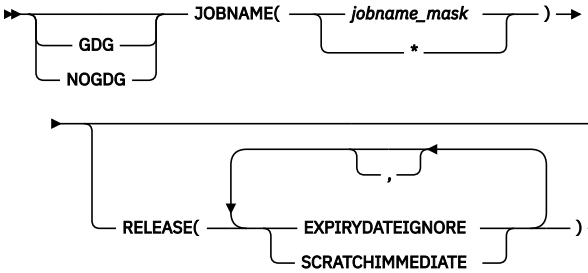




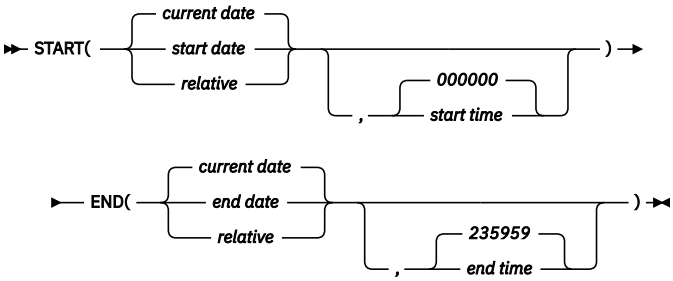
SEARCHVRS: Creating a list of vital record specifications



DSNAME OPTIONS



Date and time range



Appendix B. DFSMSrmm ISPF dialog fast path commands

Table 41 on page 513 lists the fast path commands you can use within the DFSMSrmm ISPF Dialog.

Table 41. 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 MEDINF	Display the media information definitions.
CONTROL MNTMSG	Display mount message definitions.
CONTROL OPENRULE	Display special processing for DFSMSrmm to perform during OPEN processing.
CONTROL PRTITION	Display special processing for DFSMSrmm to perform during library entry or insert, export or import, eject and CUA processing, during OPEN processing, and inventory management EXPROC processing.
CONTROL REJECT	Display volumes to be rejected.
CONTROL SECURITY	Display security classification rules.
CONTROL STATUS	Display DFSMSrmm subsystem address space status, tasks, and queued requests.
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.
DSNAME MIXED	Do not fold Data Set Names to uppercase
DSNAME UPPER	Fold Data Set Names to uppercase
EJECT Display	Show tape EJECT status.
EJECT BULK	Change tape EJECT entry to BULK.
EJECT CONVENIENCE	Change tape EJECT entry to CONVENIENCE.

© Copyright IBM Corp. 1992, 2022

Table 41. DFSMSrmm ISPF dialog fast path commands (continued)

Issue the fast path command	To
LIB	Display the Librarian Menu.
LOCAL	Display a menu your location has tailored.
OPTIONS	Display the Dialog Options Menu.
OPTIONS CONFIRM	Display the current setting for the CONFIRM command.
OPTIONS CONFIRM ON	Confirm any deletes or releases before they are processed.
OPTIONS CONFIRM OFF	Process deletes or releases without confirming.
OPTIONS DATE	Display the current date setting.
OPTIONS DATE American	Set the date format to mm/dd/yyyy.
OPTIONS DATE European	Set the date format to dd/mm/yyyy.
OPTIONS DATE ISO	Set the date format to yyyy/mm/dd.
OPTIONS DATE Julian	Set the date format to yyyy/ddd.
OPTIONS SAVE	Display the current setting for the SAVE command.
OPTIONS SAVE ON	Save commands in a data set for background processing.
OPTIONS SAVE OFF	Process commands interactively.
OPTIONS SORT	Display the Dialog Sort Options Menu.
OPTIONS SORT DATASET	Display the Data Set List Sort Options panel.
OPTIONS SORT PRODUCT	Display the Product List Sort Options panel.
OPTIONS SORT RACK	Display the Rack and Bin List Sort Options panel.
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.

Table 41. DFSMSrmm ISPF dialog fast path commands (continued)

Issue the fast path command	To
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.
TZ	Select the current time zone setting.
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.
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

Accessible publications for this product are offered through IBM Documentation (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the Contact the z/OS team web page (www.ibm.com/systems/campaignmail/z/zos/contact_z) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

© Copyright IBM Corp. 1992, 2022

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road

© Copyright IBM Corp. 1992, 2022

Poughkeepsie, NY 12601-5400
USA

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSMSrmm.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

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.

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.

abend
Abnormal end of task.

ACEE
Accessor environment element.

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*.

© Copyright IBM Corp. 1992, 2022

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.

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.

basic format
The format of a data set that has a data set name type (DSNTYPE) of BASIC. A basic format data set is a sequential data set that is specified to be neither large format nor extended format. The size of a basic format data set cannot exceed 65 535 tracks on each volume.

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.

Byte
8 bit.

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*.

CDS
Control data set.

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.

classpath
The name of a Windows-environment variable that contains the names and paths of required Java™ libraries.

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.

DASD
Direct access storage device.

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.

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 z/OS, this function is provided by DFSMS, DFSORT, and RACF. See also *system-managed storage*.

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.

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.

EAR
(1) Enterprise Application Repository. (2) Enterprise ARchive.

EDM-managed
Controlled by External Data Manager.

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. 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 written on an export list volume and then, the export operation itself 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.

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.

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.

gigabyte (GB)
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.

GMT
Greenwich mean time.

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.

hardware configuration definition (HCD)
An interactive interface in z/OS 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.

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 written on an import list volume and the stacked volumes entered, and then, the import operation itself 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 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.

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. DFSMSrmm supports large format sequential data sets for the journal.

keyword
A predefined word that is used as an identifier.

kilobyte (KB)
The information-industry meaning depends upon the context:

1. KB = 1024 (2^{10}) for real and virtual storage.
2. KB = 1000 for disk storage capacity (for example, a 4 KB fixed disk).
3. KB = 1000 for transmission rates.

large format
The format of a data set that has a data set name type (DSNTYPE) of LARGE. A large format data set has the same characteristics as a sequential (non-extended format) data set, but its size on each volume can exceed 65 535 tracks. There is no minimum size requirement for a large format data set.

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 z/OS 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 through 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.

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 z/OS 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, EHPCT, ETC, EWTC, EETC, EEWTC, EXTC, or EXWTC.

MEDIA1
Cartridge system tape.

MEDIA2
Enhanced capacity cartridge system tape.

MEDIA3
High performance cartridge tape.

MEDIA4
Extended high performance cartridge tape

MEDIA5
IBM Enterprise Tape Cartridge.

MEDIA6
IBM Enterprise WORM Tape Cartridge.

MEDIA7
IBM Enterprise Economy Tape Cartridge.

MEDIA8
IBM Enterprise Economy WORM Tape Cartridge.

MEDIA9
IBM Enterprise Extended Tape Cartridge.

MEDIA10
IBM Enterprise Extended WORM Tape Cartridge.

MEDIA11
IBM Enterprise Advanced Tape Cartridge.

MEDIA12
IBM Enterprise Advanced WORM Tape Cartridge.

MEDIA13
IBM Enterprise Advanced Economy Tape Cartridge.

megabyte (MB)
The information-industry meaning depends upon the context:

1. MB = 1 048 576 (2^{20}) for real and virtual storage.
2. MB = 1 000 000 for disk storage capacity (for example, a 4 MB fixed disk).
3. MB = 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 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.

MOF
Managed Object Format.

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.

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.

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.

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*.

petabyye (PB)
The information-industry meaning depends upon the context:

1. T = 1 125 899 906 842 624 (2^{50}) for real and virtual storage.
2. T = 1 000 000 000 000 000 for disk storage capacity (for example, 4 TB of DASD storage).
3. T = 1 000 000 000 000 000 for transmission rates.

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 z/OS 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.

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 method
The way in which DFSMSrmm retains a volume or data set before considering it for release. DFSMSrmm provides the EXPDT and VRSEL retention methods.

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 client
An instance of the DFSMSrmm subsystem running on a system that has no direct attachment to the DASD containing the DFSMSrmm control data set. The RMM client system uses TCP/IP to request the DFSMSrmm server to perform I/O to the DFSMSrmm control data set.

RMM complex (RMMplex)
One or more z/OS images that share a common DFSMSrmm control data set.

RMM server
An instance of the DFSMSrmm subsystem running on a system that has direct attachment to the DASD containing the DFSMSrmm control data set. The RMM server system uses TCP/IP to receive requests from a DFSMSrmm client to perform I/O to the DFSMSrmm control data set.

RMODE
Residence mode.

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.

SMTP
Simple Mail Transfer Protocol.

SNIA
Storage Networking Industry Association.

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 z/OS 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 z/OS task that provides services and functions to other z/OS users. Requests for service are made to the subsystem through a standard z/OS facility known as the subsystem interface (SSI). Standard z/OS 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*.

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.

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*.

terabyte (TB)
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

UDDI
Universal Description, Discovery and Integration

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.

UTC
DFSMSrmm common time support. Also known as GMT

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.

VNDR
Vendor name

VTS
Virtual tape server

VWMC
Volume write mount count

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*.

XML
eXtensible Markup Language

z/OS image
A single occurrence of the z/OS operating system that has the ability to process work.

Index

Special Characters

* (asterisk) 57
% (percent sign) 57

A

ABEND
    reserved data set name mask 59
    resetting the flag 259
    searching for data sets marked closed 389
    searching for data sets marked open 360, 361, 392
    setting the flag 254
    vital record specification 76
    volume information
        non-scratch volume 29, 211
        scratch volume 30, 211
ABEND VRS 59
accessibility
    contact IBM 517
accounting information 331, 332
action bar
    how used 9
active requests
    displaying status of 331
active requests, number of 155
ADDBIN subcommand
    overview 190
    return codes 192
    syntax 191, 467
ADDDATASET subcommand
    overview 192
    return codes 202
    syntax 193, 467
adding
    an exported logical volume 219
    bin numbers 25, 190
    bin numbers to a built-in storage location 208
    bin numbers to an installation defined location 209
    compaction information 219, 280
    creating job name 197
    data set information 42, 192
    exported logical volumes 219
    ISO/ANSI label for VOL1 229
    ISO/ANSI label version 220, 223
    logical volumes 232
    media type information 224
    mediatype information 288
    owner ID 38, 202
    owner information 38, 202
    physical volumes 219, 232
    rack number 25, 208
    recording formation information 227, 294
    scratch volumes 30
    shelf location information 190, 208
    software product information 39, 205
    vital record specification 69, 236
adding *(continued)*
    volumes 211
    volumes that reside in a manual tape library 28
adding duplicate volume serial numbers 30
ADDOWNER subcommand
    overview 202
    return codes 205
    syntax 202, 469
ADDPRODUCT subcommand
    overview 205
    return codes 207
    syntax 206, 469
ADDRACK subcommand
    overview 208
    return codes 211
    syntax 209, 469
ADDVOLUME
    syntax 475
ADDVOLUME subcommand
    overview 211
    return codes 235
    syntax 211, 470
ADDVRS subcommand
    example 106
    overview 236
    return codes 250
    syntax 237–240, 477, 478, 480
ADMIN fast path command 513
ADMIN operand 2
American date format 7
ANDVRS operand 101
applying movement and retention policies 63
archiving data 90
assigning
    a new volume serial number 289
    access to volumes 211
    scratch volume 16, 320
assistive technologies 517
authorizing users
    STGADMIN.EDG.IGNORE.TAPE 31
    STGADMIN.EDG.LISTCONTROL 131
    STGADMIN.EDG.MASTER 31
    to change volume information 31
    to ignore duplicate volume serial numbers 31
    to use LISTCONTROL subcommand 131
automated tape library
    cartridge entry processing 92
    confirming moves 95
    convenience output station 93
    ejecting volumes from 93, 315
    high capacity station 93
automatic
    data set information recording 41
    deletion of data set information 42
    owner notification 38, 117
    processing 114
    recovery 171

automatic *(continued)*
release action 115, 116
releasing of volumes 113
return to scratch status 115
volume information recording 27
volume release 113
AUTOMOVE operand 36

B

background processing 7
batch
requesting resource details in 15
batch job
scratch volume 16
submitting 12, 188
bin number
adding 25, 190
defining 25, 190
deleting 26, 305
details 129, 325
initial 26
line operators 142, 143
list 132, 351
bin numbers
adding in a storage location 190
building an export/import list 409
built-in storage locations
adding bin numbers to 190
deleting bin numbers from 305
displaying bin number details 325
BYDAYSCYCLE
retention type 74
bypass label processing
conditions when not permitted 167

C

calculating
volume retention date 82
volume usage value 348
canceling outstanding volume moves 37, 288
capacity
of volumes 21
cartridge entry processing 92
catalog date format 83
catalog retention period 75, 249
chaining
data sets in a multivolume data set 392
data sets in the same multivolume data set 361
movement policies 101
retention policies 101
chaining vital record specifications 53
CHANGEDATASET subcommand
COPYFROM 122
overview 251
return codes 264
syntax 251, 480
CHANGEOWNER subcommand
overview 265
return codes 267
syntax 265, 483
CHANGEPRODUCT subcommand
CHANGEPRODUCT subcommand *(continued)*
overview 268
return codes 269
syntax 268, 484
changes
summary of changes xxvii
CHANGEVOLUME subcommand
confirmrelease and global movement operands 484
overview 269
return codes 302
specific volume optional operands 485
syntax 271, 484, 485
CHANGEVRS
syntax 492
CHANGEVRS subcommand
overview 302
changing
data set information 251
data set information automatically 23
data set information with RMM CHANGEDATASET subcommand 42
data set name 259
expiration date 17, 118
home location with RMM CHANGEVOLUME subcommand 35
label version 283
list sort order 134
location name 287
owner information 18, 38
owner information with RMM CHANGEOWNER subcommand 265
pending status or scratch status 34
pool ID 94
release date 17, 118
retention method 50
retention period 17
shelf location 94
software product information 40, 268
vital record specification 70
volume information 31, 269
volume release date 17, 118
volume serial number 289
volume type 299
changing information for multiple volumes 31, 34
character set
chart xxii
use in statement xxii
checking DFSMSrmm status 155
clearing loan location 287
clearing the loan location
returning from a loan location 95
clearing volume location information 282
CLIST
confirming volume moves 98
confirming volume release actions 120
creating a 146
ejecting volumes 94
closed-cycle GDGs 86
command
abbreviations 187, 188
ADDBIN, TSO subcommand 190
ADDDATASET, TSO subcommand 192
ADDOWNER, TSO subcommand 202
ADDPRODUCT, TSO subcommand 205

command *(continued)*
ADDRACK, TSO subcommand 208
ADDVOLUME, TSO subcommand 211
ADDVRS, TSO subcommand 236
batch job 188
CHANGEDATASET, TSO subcommand 251
CHANGEOWNER, TSO subcommand 265
CHANGEPRODUCT, TSO subcommand 268
CHANGEVOLUME, TSO subcommand 269
CHANGEVRS, TSO subcommand 302
CONFIRM 8
DATE 7
DELETEBIN, TSO subcommand 305
DELETEDATASET, TSO subcommand 307
DELETEOWNER, TSO subcommand 309
DELETEPRODUCT, TSO subcommand 310
DELETERACK, TSO subcommand 311
DELETEVOLUME, TSO subcommand 314
DELETEVRS, TSO subcommand 318
DFSMSrmm 187
dialog
fast path commands 10
END 11
fast path 10
GETVOLUME, TSO subcommand 320
LISTBIN, TSO subcommand 325
LISTCONTROL, TSO subcommand 327
LISTDATASET, TSO subcommand 337
LISTOWNER, TSO subcommand 340
LISTPRODUCT, TSO subcommand 341
LISTRACK, TSO subcommand 342
LISTVOLUME, TSO subcommand 344
LISTVRS, TSO subcommand 349
PRINT-HI, ISPF command 15, 148
PRINT, ISPF command 15, 148
processing 7
RETURN 11
SEARCHBIN, TSO subcommand 351
SEARCHDATASET, TSO subcommand 357
SEARCHOWNER, TSO subcommand 370
SEARCHPRODUCT, TSO subcommand 373
SEARCHRACK, TSO subcommand 376
SEARCHVOLUME, TSO subcommand 380
SEARCHVRS, TSO subcommand 412
SORT 139
specifying 187
TSO subcommand 187
Command Menu 4
COMMAND operand 2
COMMANDS fast path command 513
compaction information
adding 219, 280
compression
of volumes 21
CONFIRM command 8
confirm delete option 7
confirming
all pending moves 36, 73
deletions
data set information 43
software product 41
line operators 120
list 145
manual release action 116, 119
confirming *(continued)*
using CHANGEVOLUME 36, 95
volume move 36, 95
volume release 121
console messages 152
contact
z/OS 517
container
defining 219
continuation characters
how to use xxii
CONTROL
fast path command 513
operand 2
control data set
calculating volume usage 348
control record information 327
defining resources 25
displaying details 327
system options information 131
controlling data set recording 27
controlling tape erasure 177
controlling tape initialization 177
converting from closed-cycle gdgs to scratch pools 86
COPYFROM 254
correcting data set name 259
COUNT(0)
adding with EDGPV200 panel 103
in VRS 102
creating
a CLIST 146
bin number list 132
data set list 132
dropship list 145
list of resources 132
rack number list 132
REXX exec 431
shelf location list 132
software product list 132
vital record specifications list 132
volume list 132
volume movements and actions list 146
creating a volume export list 411
creating job name
changing with RMM CHANGEDATASET subcommand 264
default used by DFSMSrmm 258
definition 76
displaying with RMM LISTDATASET subcommand 337
specifying with RMM ADDDATASET subcommand 197
CYCLES
retention type 73
cycles date format 83

D

data class recorded by DFSMSrmm 41
data set
adding information 42, 192
automatic recording of information 41
chaining data sets in a multivolume data set 392
chaining data sets in the same multivolume data set 361
changing information 42, 251
changing information automatically 23

data set *(continued)*
changing name 259
copying attributes 122
copying attributes of 123
defining 41, 192
deleting information 42, 307
details 129, 337
displaying all the data sets in a multivolume set 23
expiration date 46
fully qualified name 56
generic data set name 56
how to define data set name 56
LASTREF extra days 47
line operators 142, 143
list 132, 357
list support 148
management class 337
marking incorrect 254
name mask 244
non-restricted operands or fields 251
restricted operands or fields 251
retain by job name 63
retaining based on catalog status 74
retaining by job name 76
retaining until expired 74, 75
retaining while cataloged 74
retaining while open 75
retention date 82
retention method 125
saved commands 11
searching for data sets in a multivolume data set 361, 392
searching for data sets marked closed 389
searching for data sets marked open 360
sequence number 42, 130
target
retention of using EDG_EXIT100 125
uncataloging 43
updating information 42, 251
updating information automatically 23
vital record specification management value 337
data set name filter
separating from policy 102
data set recording 27
data set vital record specification
adding 70
data set name mask 53
definition 53
matching data set names 63
specifying data set names 55
data sets
adding information about 192
changing information about 251
excluding from VRSEL processing 255, 263, 368
identifying for retention 55
retaining 19
retaining while cataloged 201, 263
DATASET
fast path command 513
operand 2
DATE command 7
date format
catalog date 83
cycles date 83
date format *(continued)*
setting for a session 7
used in initialization messages 161
DAYS
retention type 74
default sort order
changing 135
defining release policies 79
defining vital record specification
a creating job name 197
a scratch volume 27
data set 41
exported logical volumes 219
home location 88
library names 88
loan location 94
logical volumes 232
master volume 29
movement policies 88
original expiration date 118
owner 38, 202
physical volumes 232
resource 25
retention policies 73
scratch volume 30
shelf location 25, 190, 208
software product 39, 205
user volume 29
vital record specification 69
volume 27, 211
with RMM ADDBIN subcommand 190
with RMM ADDPRODUCT subcommand 205
with RMM ADDRACK subcommand 208
with RMM ADDVOLUME subcommand 29, 211, 219, 232
with RMM ADDVRS subcommand 25, 27, 30, 38, 39, 41, 69, 73, 88, 94, 118, 197, 202
DELETEBIN subcommand
overview 305
return codes 306
syntax 305, 492
DELETED vital record specification 76
DELETED VRS 60
DELETEDATASET subcommand
overview 307
return codes 308
syntax 307, 492
DELETEOWNER subcommand
overview 309
return codes 310
syntax 309, 492
DELETEPRODUCT subcommand
overview 310
return codes 311
syntax 310, 493
DELETERACK subcommand
overview 311
return codes 313
syntax 312, 493
DELETEVOLUME subcommand
overview 314
return codes 317
syntax 315, 493
DELETEVRS subcommand
overview 318

DELETEVRS subcommand *(continued)*
return codes 319
syntax 318, 494
deleting
an empty rack with a media name that does not match VLPOOL 313
bin number 305
bin numbers from a built-in storage location 312
bin numbers from an installation defined location 312
confirm
data set information 43
software product information 41
owner ID 309
owner information 309
rack number 311
shelf location information 311
software product information 310
vital record specification 70
delimiters xxii
detection
of duplicate generation data sets 66
DFSMSdfp labeling support 222, 286
DFSMShsm
backup and migration tapes 249
migration tapes, moving 90
retaining DFSMShsm ABARS backups 249
DFSMSrmm
displaying status of 331
getting started 1
ISPF dialog 1
navigating through DFSMSrmm
using the DFSMSrmm panels 9
navigating through the DFSMSrmm dialog 9
reason codes 423
refreshing the DFSMSrmm installation exits 155
removing DFSMSrmm from the system 159
resources 25
return codes 423
starting 1
stopping 11
tape initialization and erase control 177
TSO subcommands 187
wrong label processing 178
DFSMSrmm installation exits
managing 186
DFSMSrmm ISPF exec
operands 2
using 2
DFSMSrmm processing options
changing 6
DFSMSrmm TSO subcommands 467
diagnosing errors 3, 115
dialog
creating CLISTS of executable commands 146
exiting 11
options 6
starting 1
Dialog Options Menu
using 8
disaster recovery 53
displaying
all the data sets in a multivolume set 23
bin number details 325, 342
control record information 327
displaying *(continued)*
data set details 337
DFSMSrmm control data set information 131
disposition control options 131
installation options 131
owner details 340
parmlib options information 327
rack number details 342
required destination 403
shelf location details 325, 342
software product details 341
vital record specification details 302, 349
volume details 344
disposition control
default sticky label description 185
displaying options 131
sticky label support 185
system options information 331, 332
DSNAME
fast path command 513
DSNAME case
enabling lowercase or mixed case 8
dummy tag
ABEND 76
matching to data set names 63
duplicate generation data sets
detection of 66
duplicate volume serial number
adding 30
adding with RMM ADDVOLUME subcommand 211
ignoring with EDG_EXIT100 30
managing 40
redefining 30
searching for duplicate volumes 408

E

EDG_EXIT100 installation exit
copying data set attributes with 123
ignoring duplicate volume serial number with 30
retaining target data set 125
EDG0001I 161
EDG0101I 161
EDG0103D 162
EDG0104E 162
EDG0105I 161
EDG0110D 161
EDG0114I 161
EDG0115D 162
EDG0117D 162
EDG0123D 162
EDG0154I 171
EDG019VM installation exit 166
EDG0204I 161
EDG0215D 162
EDG0228E 162
EDG0238E 163
EDG0239E 163
EDG0240E 163
EDG0241E 163
EDG0242E 163
EDG0243E 163
EDG0353I 163
EDG0358D 163

EDG0361D 164
EDG1001A 162
EDG1101I 164
EDG1105I 164
EDG1106I 164
EDG1107D 164
EDG2103D 171
EDG2104E 171
EDG2107E 171
EDG2108E 171
EDG2110I 171
EDG2111I 171
EDG2112I 171
EDG2113I 171
EDG2114I 171
EDG2115I 171
EDG2116A 171
EDG4000D 169
EDG4001D 169
EDG4005E 170
EDG4006E 170
EDG4007E 170
EDG4008A 170
EDG4010D 170
EDG4012D 170
EDG4013I 165
EDG4021I 167
EDG4023I 167
EDG4024I 167
EDG4025I 168
EDG4026I 168
EDG4027I 168
EDG4028I 168
EDG4032I 168
EDG4033I 168
EDG4035I 168
EDG4036I 168
EDG4041I 168
EDG4048I 169
EDG4049I 169
EDG4058I 169
EDG4059I 169
EDG4060I 169
EDG4064D 170
EDG4065I 169
EDG4066I 169
EDG4067I 169
EDG6627A 165
EDG6628A 165
EDG6663D 165
EDG9115I 164, 172
EDG9116I 172
EDGINERS utility
  automatic processing 116
  manual processing 172
  using EDGINERS 172
  wrong label processing 178
EDGLABEL procedure
  description 172
  message responses 178
EDGPV200 panel
  adding COUNT(0) with 103
EDGRPD34 exec 148, 149
EDGRPTD utility 129
EDGRVCLN REXX EXEC 110
EDGV01D 167
ejecting
  cartridges to an output station 8
  volumes by building a list of DFSMSrmm subcommands 94
  volumes from tape libraries 93
  volumes with ISMF or z/OS console command 302
  volumes with RMM DELETEVOLUME subcommand 315
elapsed days retention type 74
empty rack or bin number 26
END command 11
ending a search for volume information 395
ERASE request format 181
erase volume release action 18, 116
erasing volumes 116, 172
error diagnosis 3, 115
European date format 7
example
  using TSO help 188
examples
  adding
    a data set with RMM ADDVOLUME subcommand 201
    a duplicate volume with RMM ADDVOLUME subcommand 211
    a new owner with RMM ADDOWNER subcommand 205
    a new product with RMM ADDPRODUCT subcommand 207
    empty bin to the storage location with RMM ADDBIN subcommand 192
    empty bins to a storage location with RMM ADDRACK subcommand 208
    empty racks to the removable media library with RMM ADDRACK subcommand 208
    tape cartridge with RMM ADDVOLUME subcommand 235
    volumes in a system-managed library with RMM ADDVOLUME subcommand 235
    volumes using status information from the TCDB 235
    WORM volumes TCDB 235
  associating
    a volume to a software product with RMM CHANGEVOLUME subcommand 302
  building
    a list of eject commands with RMM SEARCHVOLUME subcommand 408
  changing
    owner information with RMM CHANGEOWNER subcommand 267
    owner of a software product with RMM CHANGEPRODUCT subcommand 269
    the MANAGEMENTVALUE assigned to a data set 264
  confirming
    volume actions with RMM CHANGEVOLUME subcommand 301
  creating
    a list of bin numbers using the CONTINUE operand with RMM SEARCHBIN subcommand 355
    a list of bin numbers with RMM SEARCHBIN subcommand 355

examples *(continued)*
creating *(continued)*
a list of data sets with RMM SEARCHDATASET subcommand 369
a list of duplicate volumes with RMM SEARCHVOLUME subcommand 408
a list of empty or available bin numbers with RMM SEARCHBIN subcommand 354
a list of owners with RMM SEARCHOWNER subcommand 373
a list of scratch volumes with RMM SEARCHRACK subcommand 379
a list of software products with RMM SEARCHPRODUCT subcommand 376
a list of volumes that belong to an owner with RMM SEARCHVOLUME subcommand 408
defining
vital record specification 249
with RMM ADDVRS subcommand 249
deleting
a data set vital record specification 319
a volume vital record specification with RMM DELETEVRS subcommand 319
information about a volume with RMM DELETEVOLUME subcommand 317
information about data sets with RMM DELETEDATASET subcommand 308
software product information with RMM DELETEPRODUCT subcommand 311
user ID and node for an owner with RMM CHANGEOWNER subcommand 267
user IDs with RMM CHANGEVOLUME subcommand 301
displaying
defaults table entries with RMM LISTCONTROL subcommand 336
information about DFSMSrmm subsystem, subsystem requests, and task status 330
information for a single data set with RMM LISTDATASET subcommand 338
installation's options and rules with RMM LISTCONTROL subcommand 331
location definitions with RMM LISTCONTROL subcommand 334
media information with RMM LISTCONTROL subcommand 334
security class masks with RMM LISTCONTROL subcommand 334
JCL to run a batch TSO TMP 188
listing
vital record specifications with RMM SEARCHVRS subcommand 420
managing
special dates with RMM ADDVRS subcommand 249
marking
a volume as no longer in transit with RMM CHANGEVOLUME subcommand 302
moving
a volume to an automated tape library with RMM CHANGEVOLUME subcommand 301
releasing
volumes associated with a software product with RMM DELETEPRODUCT subcommand 311
removing
examples *(continued)*
removing *(continued)*
bin numbers with RMM DELETEBIN subcommand 306
shelf location information with RMM DELETEBIN subcommand 306
shelf location information with RMM DELETERACK subcommand 313
requesting
a volume from the default scratch pool with RMM GETVOLUME subcommand 324
information a shelf location with RMM LISTBIN subcommand 326
information a shelf location with RMM LISTRACK subcommand 343
information about a vital record specification with RMM LISTVRS subcommand 350
owner information with RMM LISTOWNER subcommand 340
software product information with RMM LISTPRODUCT subcommand 342
volume serial information with RMM LISTVOLUME subcommand 346
volume statistics with RMM LISTVOLUME subcommand 347
retaining
DFSMShsm ABARS backups 249
DFSMShsm backup and migration tapes 249
with RMM ADDVRS subcommand 249
retention by cycles 55
setting
a creating job name 264
testing
return codes in a CLIST 423
return codes in a REXX procedure 423
transferring
volume ownership with RMM DELETEOWNER subcommand 309
updating
data set information with RMM CHANGEDATASET subcommand 264
exec
DFSMSrmm ISPF 2
REXX 431
Exit Menu 11
exiting the DFSMSrmm dialog 11
exits
managing installation exits 186
EXPDT managed
source data set
retention of 126
EXPDT retention method
candidates for 49
changing to 50
compared to VRSEL retention method 48
introduction 45
RETAINBY attribute 45
using 19
expiration
based on last reference 47
date 17, 45, 118
JCL date 77
processing 113
expiration date protection 77, 118

expirydateignore 77, 79, 118
exported logical volume
  adding 219
extra days retention type 74
extract file processing 22
EXTRADAYS
  retention type 74
EXTRADAYS example 106

F

F line command 30
fast path commands
  issuing 10
feedback xxv
fields, restricted and non-restricted 33, 251
filtering with data set name masks 63
foreground or background processing
  option for 8
foreground processing 7
freeing up space in system-managed libraries 88
fully qualified data set name 56
function menus 4

G

general user
  getting started with 1
generation data group (GDG)
  base names 66
  closed-cycle 86
  converting to use catalog control 87
  converting to use scratch 87
generation data sets
  detection of duplicate 66
generic data set
  description 56
  using 244
generic data set name 56
generic job name description 58
generic volume serial number
  retaining volumes by 69
getting started 1
GETVOLUME subcommand
  overview 320
  return codes 324
  syntax 320, 495
global confirmation
  examples 97
  performing 36
  undoing movement and action confirmation 36
glossary 523
grouping volumes 20

H

handling volume discrepancies 184
help
  online RMM TSO subcommand help 188
  requesting 11
home location
  changing 88
  changing with CHANGEVOLUME subcommand 35
home location *(continued)*
  setting 88, 223
how to abbreviate
  commands xxii
  operands xxii

I

I line command 148
I/O errors on a volume 115
IEF233A 166
IFG019VM installation exit 166
ignoring duplicate volume serial numbers 30
ignoring expiration date 77, 113
implementing a data archive process 90
in transit 95
information about your resources 5
information recording performed by DFSMSrmm 41
INIT request format 181
initial
  rack or bin number 26
  volume serial number 30
initialization messages 161
initialize volume release action 18, 116
initializing volumes 116, 172
installation defined storage locations
  adding bin numbers to 190
  deleting bin numbers from 305
  displaying bin number details 325
  obtaining information about 26
installation exits
  EDGUX100 30
  managing 186
inventory management
  confirming volume movements 95
  report 129
ISMF selection menu
  requesting RMM dialog from 1
ISO date format 7
ISO/ANSI
  changing label version 283
  changing the label version 296
  defining label version 220, 223
  defining label version in the VOL1 label 229
  labeling 172
  replacing the label version 296
  searching for ISO/ANSI label information 395
  Version 3 174
  Version 4 174
ISPF primary option menu
  requesting RMM dialog from 2

J

JCL expiration date 77
job name
  controlling policy selection 63
  creating job name definition 76
  generic job name 58
  how to define 58
  retaining data sets by job name 76
  using with ADDDATASET subcommand 193

job name mask 58
jobname
used to apply policies 62
journal
displaying status of 331
operator system error messages 171
REXX variable 449
system options information 131, 331, 332
Julian date format 7

K

keyboard
navigation 517
PF keys 517
shortcut keys 517

L

label
scanning a tape volume label 184
LABEL procedure
description 172
labeling ISO/ANSI tape labels 172
last referenced retention type 74
LASTREF extra days 47
LASTREFERENCEDAYS
retention type 74
LIBRARIAN fast path command 513
LIBRARIAN operand 2
limit, search 132
limiting the search for volume information 395
line operators
confirm 120, 146
data set 142
rack and bin number 142
release volumes 121, 122
shelf location 142
software product 142
using 142
vital record specification 142
volume 142
LISTBIN subcommand
overview 325
return codes 326
syntax 325, 496
LISTCONTROL subcommand
overview 327
return codes 336
syntax 327, 496
LISTDATASET subcommand
overview 337
return codes 339
syntax 338, 497
LISTOWNER subcommand
overview 340
return codes 340
syntax 340, 497
LISTPRODUCT subcommand
overview 341
return codes 342
syntax 341, 497
LISTRACK subcommand
LISTRACK subcommand *(continued)*
overview 342
return codes 344
syntax 343, 497
lists
bin number 351, 376
changing sort order 134
CLISTS of executable commands 146
confirm volume moves 145
creating 132
data set 132, 357
dropship 145
printing 15, 148
rack number 376
requesting 132
scratch volume 143
search limit 132
shelf location 351, 376
software product 132, 373
sort options for 13
sorting 13, 134
using 12, 132
vital record specification 412
volume 132, 380
LISTVOLUME subcommand
overview 344
return codes 348
syntax 345, 497
LISTVRS subcommand
overview 349
return codes 304, 350
syntax 303, 349, 498
loan location
changing 299
clearing loan location 287
defining 219
moving to a 94
returning from a 95
LOCAL fast path command 513
LOCAL operand 2
location
changing 287
location definitions 334
LOCDEF command 90
logical volume, changing 299

M

M line command 56, 148
management class
defining management and retention policy 236
displaying data set information 337
recorded by DFSMSrmm 41
specifying 78, 243
managing
data sets based on data set name 76
data sets closed by ABEND processing 76
data sets left open by a system failure 75
data sets that have deleted flag set on 76
DFSMSrmm installation exits 186
multiple data sets with data set masks 63
special dates 249
vital records 73
volumes with the same volume serial number 30

managing SEARCH
large results list 130
manual move control 90
manual processing 114
manual tape library
adding volumes that are resident in 28
moving volumes to 92
MANUALMOVE operand 36
marking a stacked volume closed 280
marking DFSMSrmm—managed data sets incorrect before restarting jobs 254
mask, data set name 244
masking characters 67
master status volumes, controlling tape initialization 177
master volume
definition 29
overwriting 27
searching for information about 368
matching order for vital record specifications 63
MATCHVRS primary command 56
maximum retention period 77
media information 334
media names 28
media types 28
mediatype information
adding 224, 288
menu
Command 4
Exit 11
primary option menu 1
user 3
messages
displaying case 331, 332
EDG0001I 161
EDG0101I 161
EDG0103D 162
EDG0104E 162
EDG0105I 161
EDG0110D 161
EDG0114I 161
EDG0115D 162
EDG0117D 162
EDG0123D 162
EDG0154I 171
EDG0204I 161
EDG0215D 162
EDG0228E 162
EDG0238E 163
EDG0239E 163
EDG0240E 163
EDG0241E 163
EDG0242E 163
EDG0243E 163
EDG0353I 163
EDG0358D 163
EDG0361D 164
EDG1001A 162
EDG1101I 164
EDG1105I 164
EDG1106I 164
EDG1107D 164
EDG2103D 171
EDG2104E 171
EDG2107E 171
messages *(continued)*
EDG2108E 171
EDG2110I 171
EDG2111I 171
EDG2112I 171
EDG2113I 171
EDG2114I 171
EDG2115I 171
EDG2116A 171
EDG4000D 169
EDG4001D 169
EDG4005E 170
EDG4006E 170
EDG4007E 170
EDG4008A 170
EDG4010D 170
EDG4012D 170
EDG4013I 165
EDG4021I 167
EDG4023I 167
EDG4024I 167
EDG4025I 168
EDG4026I 168
EDG4027I 168
EDG4028I 168
EDG4032I 168
EDG4033I 168
EDG4035I 168
EDG4036I 168
EDG4041I 168
EDG4048I 169
EDG4049I 169
EDG4058I 169
EDG4059I 169
EDG4060I 169
EDG4064D 170
EDG4065I 169
EDG4066I 169
EDG4067I 169
EDG6627A 165
EDG6628A 165
EDG6663D 165
EDG9115I 164, 172
EDG9116I 172
EDGV01D 167
IEF233A 166
initialization 161
operator 160
system error 171
tape mount 165
tape processing 169
tape validation 167
minimum retention with WHILECATALOG 75, 249
MODIFY command 152
mount message processing 165
mouse button 9
movement policies
chaining 101
defining with VRS 99
moving
DFSMShsm migration tapes 90
from storage locations 95
list 146
out of system-managed libraries 88

moving *(continued)*
to loan locations 94
volumes 73, 94
volumes using manual move control 90
multivolume chain
last volume in a multivolume chain 301
multivolume sets
retention of 45

N

name filtering rules 61
name vital record specification
adding 70, 237
definition 53
example 55
name, data set 244
navigating through DFSMSrmm
using fast path commands 10
using the action bar 9
without using the action bar 9
navigation
keyboard 517
new
summary of changes xxvii
NEXTVRS example 106
NEXTVRS operand 101
no label
DFSMSrmm support 182
obtaining information about 166
non-GDG data sets
retaining cycles of 105
non-restricted operands or fields 33, 251
non-restricted volume information 32
non-scratch volume
adding 29
notational conventions xix
notification of volume release 19
notify owner release action 18, 117
notifying owners 38
number of cycles or copies retention type 73, 74

O

obtaining
information about NL tapes 166
information about your resources 5
parmlib suffix 331, 332
online help 188
OPEN
resetting the closed flag 280
resetting the open flag 292
searching for volumes marked open 400
OPEN vital record specification 75
OPEN VRS 61
operands, restricted and non-restricted 33, 251
operator
console 152
initialization message responses 161
LABEL procedure message responses 178
messages 160
procedures 151
procedures, operator 151
operator *(continued)*
sticky label support 185
system error message responses 171
tape mount message responses 165
tape processing message responses 169
tape validation message responses 167
options
changing 6
confirm delete 7
date format 7
dialog 6
DSNAME case 8
ejecting cartridges 8
foreground or background processing 8
reusing variable information 8
OPTIONS fast path command 513
overriding Automatic Move Processing 36
overriding expiration date 20, 77
overwriting master volumes 27
owner
adding information 202
changing information 265
defining 202
deleting information 309
details 129
ID
adding 202
changing 265
defining 202
deleting 309
details 129
searching 371
notification 117
requesting volume return 116
searching information 371
updating information 265
OWNER fast path command 513
owner ID
adding 38
changing 38
deleting 39
displaying information about 340
owner IDs
defining 38
owner information
changing 18
OWNER operand 2
owners
adding information about 202
changing information about 265

P

parmlib options
displaying details 327
implementing new options 155
pending movement and action
confirming 36, 73
confirming using the DFSMSrmm ISPF Dialog 96
for return to scratch volumes 129
permanent I/O errors 115
permanent retention with 99365, 99366, 1999/365, 1999/366 77
physical volume

physical volume *(continued)*
changing 299
point-and-shoot fields 9
policies
using jobname to apply 62
policy
movement 73
retention 73
separating from data set name filter 102
setting up 20
pool
adding rack number 25, 26
changing information for a volume 94
details 146
moving volumes 94
viewing pool ids 29
pooling requirements 20
preventing a stacked volume from moving 292
primary option menu 1
priming DFSMSrmm ISPF dialog panels 8
PRINT command 15, 148
PRINT-HI command 15, 148
printing lists and details panels 15, 148
private volume
overwriting master volumes 27
searching for information about 368
problem determination aid
controlling tracing 160
initialization messages 164
system error messages 172
problem diagnosis 3
processing
background 7
batch, commands in 12
foreground 7
saved commands data set 11
processing NL tapes 166
PRODUCT fast path command 513
PRODUCT operand 2
programming interface information 522
pseudo-GDG
definition 67
specifying 244
pulling scratch volumes 143

Q

queued requests
displaying status of 331
quiescing DFSMSrmm 159

R

RACF
STGADMIN.EDG.IGNORE.TAPE 31
STGADMIN.EDG.LISTCONTROL 131, 327
STGADMIN.EDG.MASTER 31
STGADMIN.EDG.RELEASE 314
RACK fast path command 513
rack number
adding 25, 208
adding to a pool 26
defining 25, 208
rack number *(continued)*
deleting 26, 311
deleting an empty rack with a media name that does not match VLPOOL 313
details 129, 342
initial 26
line operators 142, 143
list 132, 376
removing 291
RACK operand 2
rack pool
setting up 25
viewing information about 26
ready to scratch
confirming status 282
movement report 129
NOTREADYTOSCRATCH operand 281
READYTOSCRATCH operand 281
using to confirm volume movement 36, 98
reason codes 424
reclaiming volumes from pending status or scratch status 34
Recommendation
media type 289
recording format 228
year and day format 294
recording
data set information 23, 41
volume information
changing information 17
recording format information
adding 227, 294
recording format names 28
redefining duplicate volume serial numbers 30
refreshing DFSMSrmm 155
relabeling a volume 116
release action
confirming 116, 119
list 146
setting 114
setting a 18
release date 17
release notification 19
release processing 114
releasing a volume
associated with a software product 41
confirming 121
DELETEVOLUME subcommand 314
description 113
line operators 121, 122
manually 18, 120
notifying owner 38, 117
release options 118
types of 18
releasing volumes 113
removable media library
moving volumes 73
pools 25
shelf locations 25
shelves 25
removing
DFSMSrmm from the system 159
last volume in a multivolume chain 301
rack number 291

removing *(continued)*
shelf location information with RMM DELETEBIN subcommand 306
shelf location information with RMM DELETERACK subcommand 313
single volume from the end of a multivolume chain. 291
replace volume release action 18, 115
replacing
volumes with duplicate volume serial numbers 40
volumes with permanent errors 115
report
extract data set 129
inventory 129
requesting
bin number details 129, 325
bin number list 132, 351
confirm delete option 7
control record information 327
data set details 129, 337
data set list 132, 357
dropship list 145
information 129
list of resources 132
lists 129
owner details 129, 340
parmlib options information 327
pool details 146
rack number details 129, 342
rack number list 132, 376
scratch volume 16, 113, 320
shelf location details 129, 325
shelf location list 132, 351
software product details 129, 341
software product list 132, 373
vital record specification details 129, 302, 349
vital record specification list 132, 412
volume details 129, 344
volume list 132, 380
volume movements and actions list 146
volume release 19
requesting help 11
requesting resource details
in batch 15
requesting volumes 113
required destination 403
required product knowledge xix
reserved words
ABEND 76
DELETED 76
OPEN 75
resetting in transit flag 302
resetting the ABEND flag 290
resource
defining 25
information 129
lists 132
system options 131
restarting DFSMSrmm subsystem 151, 155
restricted operands or fields 33, 251
Restrictions
3590 media 117
adding volumes in a multivolume data set 227
CHANGEVOLUME subcommand 34
DELETEOWNER 39
Restrictions *(continued)*
deleting a data set 42, 307
owner information 38
pseudo-GDG vital record specification 68
RETAINBY
for EXPDT retention method 45
for VRSEL retention method 45
of multivolume sets 45
retaining
based on catalog status 74
by elapsed days 74
by extra days 74
by job name 76
by last referenced date 74
by number of cycles or copies 73, 74
data sets that have 'deleted' flag set on
when closed during ABEND processing 75
while open 75
target data set
using EDG_EXIT100 125
until expired 74, 75
using dialog to specify data set retention 56
volumes
using GDG name 66
with special dates 59
volumes with special dates 78
while cataloged 74
while closed during ABEND processing 75
with BYDAYSCYCLE 74
with CYCLES 73
with DAYS 74
with EXTRADAYS 74
with LASTREFERENCEDAYS 74
with UNTILEXPIRED 75
with WHILECATALOG 74
retaining a data set using a vital record specification management value 107
retaining cycles of non-GDG data sets 105
retaining data sets as long as they are cataloged 108
retaining data sets based on expiration date 108
retaining data sets based on the last referenced date 106
retaining generation data group data sets 107
retaining volumes
by generic volume serial number 69
by specific volume serial number 69
retention
identifying data sets for 55
of multivolume sets 45
source data set
EXPDT managed 126
VRSEL managed 125
types of 73
retention date
calculating 80
for data sets
obtaining information 367
obtaining information 404
retention method
changing 125
changing to EXPDT 50
deciding between EXPDT and VRSEL 48
EXPDT 19, 125
expiration date 45
for data set 125

retention method *(continued)*
introduction 45
tasks for 47
VRSEL 20
retention period 77, 118
retention policies
chaining 101
defining with VRS 99
return codes
ADDBIN subcommand 192
ADDDATASET subcommand 202
ADDOWNER subcommand 205
ADDPRODUCT subcommand 207
ADDRACK subcommand 208
ADDVOLUME subcommand 235
ADDVRS subcommand 250
CHANGEDATASET subcommand 264
CHANGEOWNER subcommand 267
CHANGEPRODUCT subcommand 269
CHANGEVOLUME subcommand 302
CHANGEVRS subcommand 304
DELETEBIN subcommand 306
DELETEDATASET subcommand 308
DELETEOWNER subcommand 310
DELETEPRODUCT subcommand 311
DELETERACK subcommand 313
DELETEVOLUME subcommand 317
DELETEVRS subcommand 319
description 423
GETVOLUME subcommand 324
LISTCONTROL subcommand 336
LISTDATASET subcommand 339
LISTOWNER subcommand 340
LISTPRODUCT subcommand 342
LISTRACK subcommand 326, 344
LISTVOLUME subcommand 348
LISTVRS subcommand 350
SEARCHBIN subcommand 356
SEARCHDATASET subcommand 370
SEARCHOWNER subcommand 373
SEARCHPRODUCT subcommand 376
SEARCHRACK subcommand 380
SEARCHVOLUME subcommand 412
SEARCHVRS subcommand 421
RETURN command 11
return to owner release action 18, 116
return to scratch release action 18, 115
returning a volume to scratch status 79
REUSASID=YES
on START command 152
reversing movement and action confirmation 36
REXX exec
creating 431
testing return codes 423
using TRACE in 3
variables used 431
RMM dialog
requesting from ISMF selection menu 1
requesting from ISPF primary option menu 2
requesting from RMMISPF exec 2
RMM TSO command
issuing 187
RMM TSO subcommands
issuing 187
RMM TSO subcommands *(continued)*
online help 188
using with System REXX 189
RMMISPF exec
requesting RMM dialog from 2

S

SCAN request format 184
scanning
volumes not defined to DFSMSrmm 27
scanning a tape volume label 184
scanning labels on volumes 172
scratch pool
converting from closed-cycle gdgs 86
volume
adding 30
assigning 16, 113
defining 27
list 143
requesting 16, 113
scratch volume
assigning 320
requesting 320
searching for information about 368
scratchimmediate 79, 118
search limit 132
SEARCHBIN subcommand
overview 351
return codes 356
syntax 352, 498
SEARCHDATASET subcommand
overview 357
return codes 370
syntax 358, 498
searching
for data sets in the same multivolume data set 361, 392
for data sets marked closed 389
for data sets marked open 360
for volumes marked open 400
owner information 370
SEARCHOWNER subcommand
overview 370
return codes 373
searching information
owner information 370
syntax 371, 501
SEARCHPRODUCT subcommand
overview 373
return codes 376
syntax 374, 502
SEARCHRACK subcommand
overview 376
return codes 380
syntax 377, 502
SEARCHVOLUME subcommand
overview 380
return codes 412
syntax 382, 502
using 143
SEARCHVRS
syntax 412
SEARCHVRS subcommand
overview 412

SEARCHVRS subcommand *(continued)*
return codes 421
syntax 413, 510
security class masks 334
SELECT primary command 13
selecting command processing 7
sending to IBM
reader comments xxv
separating data set name filter from policy 102
sequence number 42, 130
session, DFSMSrmm
starting 1
setting
date format 7
expiration date 77
home location 88
owner notification 38
policy requirements 20
setting a default vital record specification 57
time zone 7
shelf location
adding information 25, 190, 208
changing 94
defining 25, 190, 208
deleting information 26, 305, 311
details 129, 325, 342
line operators 142, 143
list 132, 351, 376
shortcut keys 517
shutting down the DFSMSrmm subsystem
operator system error message issued 171
process for 158
software product
adding information 39, 205
changing information 40, 268
defining 39, 205
deleting information 41, 310
details 129, 341
line operators 142, 143
list 132, 373
updating information 40, 268
volume 40, 41
software products
adding information about 205
changing information about 268
SORT command
to sort a list 13
using 139
SORT options panels
using 13
sort order
changing 135
sorting
default sort order 14
directions 14
lists 134
options 134
order 134
priorities 14
source data set
managed by EXPDT
retaining 126
managed by VRSEL
retaining 125
special dates, support for 78
stacked physical volume, changing 299
START command
REUSASID=YES operand 152
starting
DFSMSrmm dialog 1
DFSMSrmm subsystem 151
starting a search for volume information 395
status for a volume 27
STGADMIN.EDG.IGNORE.TAPE 31
STGADMIN.EDG.LISTCONTROL 131, 327
STGADMIN.EDG.MASTER 31
STGADMIN.EDG.RELEASE 314
sticky label
default label description 185
processing 185
stopping
DFSMSrmm dialog 11
DFSMSrmm subsystem 158
session, DFSMSrmm
stopping 11
storage class recorded by DFSMSrmm 41
storage group recorded by DFSMSrmm 41
storage location
adding bin number 25
deleting bin number 26
moving from 95
moving volumes 73
shelf locations 25
submitting
batch job 12, 188
summary of changes.
V2R4 xxvii
V2R5 xxvii
SUPPORT fast path command 513
support for OPC restart capability to mark DFSMSrmm-managed data sets incorrect before restarting jobs 254
SUPPORT operand 2
syntax
SCAN request 184
syntax diagrams
how to read xix
system error messages 171
system options 131
System REXX
using with RMM TSO subcommands 189
system-managed library
adding volumes example 235
ejecting volumes 93, 408
freeing up space 88
moving volumes 88
moving volumes to 92
obtaining information about resident volumes 27

T

tape configuration data base
obtaining information from 27
updating volume information 92
validating library name 92
tape data set
retention method 45
tape label
expiration date 118

tape label *(continued)*
scanning 184
specifying 223, 397
type 223, 397
validation 92
tape librarian
deleting an owner ID 39
tape mount messages 165
tape processing messages 169
tape validation messages 167
tape volume
erasing 181
erasing manually 172
initializing 181
initializing manually 172
list support 149
scanning manually 172
tapes
tasks associated with 22
using 22
target data set
retention of using EDG_EXIT100 125
tasks
displaying status of 331
testing
return codes in a CLIST 423
return codes in a REXX procedure 423
time zone
considerations 149
setting 7
TRACE operand 2
tracking I/0 errors 115
trademarks 522
transferring volume ownership 39, 309
TSO subcommand
abbreviations 187, 188
ADDBIN subcommand 190
ADDDATASET subcommand 192
ADDOWNER subcommand 202
ADDPRODUCT subcommand 205
ADDRACK subcommand 208
ADDVOLUME subcommand 211
ADDVRS subcommand 236
batch job 188
CHANGEDATASET subcommand 251
CHANGEOWNER subcommand 265
CHANGEPRODUCT subcommand 268
CHANGEVOLUME subcommand 269
CHANGEVRS subcommand 302
DELETEBIN subcommand 305
DELETEDATASET subcommand 307
DELETEOWNER subcommand 309
DELETEPRODUCT subcommand 310
DELETERACK subcommand 311
DELETEVOLUME subcommand 314
DELETEVRS subcommand 318
GETVOLUME subcommand 320
issuing 187
issuing from MODIFY command 154
LISTBIN subcommand 325
LISTCONTROL subcommand 327
LISTDATASET subcommand 337
LISTOWNER subcommand 340
LISTPRODUCT subcommand 341
TSO subcommand *(continued)*
LISTRACK subcommand 342
LISTVOLUME subcommand 344
LISTVRS subcommand 349
online help 188
return codes 423
SEARCHBIN subcommand 351
SEARCHDATASET subcommand 357
SEARCHOWNER subcommand 370
SEARCHPRODUCT subcommand 373
SEARCHRACK subcommand 376
SEARCHVOLUME subcommand 380
SEARCHVRS subcommand 412
specifying 187
using 187
using from a REXX exec 189
using from a TSO CLIST 189
using from an operator console 188
using with System REXX 189
variables 442
TSO subcommands
variables by subcommand 432
types of retention 73
TZ fast path command 513

U

uncataloging a data set 43
undoing movement and action confirmation 36
until expired
closed data sets retained 75
data sets retained until expired 74, 75
data sets retained while open 75
relationship with COUNT and WHILECATALOG operands 75
retention by open status 75
retention by volume expiration date 75
until expired retention type 75
UNTILEXPIRED
retention type 75
updating
data set information automatically 23
data set information with RMM CHANGEDATASET subcommand 42, 251
owner information 18, 38, 265
retention method 50
software product information 40, 268
vital record specification 70
volume information 17, 31, 269
updating vital record specifications 110
usage
of volumes 21
USER fast path command 513
user interface
ISPF 517
TSO/E 517
user menus 3
USER operand 2
user status volumes, controlling tape initialization 177
user volume
definition 29
overwriting 27
searching for information about 368
user, general 1

using the action bar 9
using the EDG_EXIT100 installation exit 30
using the EXPDT retention method 19
using the Volume Search panel 144
using the VRSEL retention method 20

V

validating
  library name 92
variables, TSO subcommand 442
virtual tape server
  adding an exported logical volume 219
  building an export/import list 409
  changing logical volume location 282
  changing volume type 299
  ejecting a logical volume 284
  identifying logical volumes to DFSMSrmm 409
vital record group 55
vital record management
  ABEND 76
  based on data set name 76
  chaining 53
  changing 70
  changing information 302
  creating list 132, 412
  defining 69, 236
  DELETED 76
  deleting 70, 318
  line operators 142, 143
  linking 236
  managing vital records 73
  matching data set names 63
  matching job names 63
  matching order 63
  matching types 337
  OPEN 75
  policy requirements 20
  requesting information 129, 349
  retaining DFSMShsm ABARS backups 249
  retaining DFSMShsm backup and migration tapes 249
  setting an installation default 57
  updating 70
  updating vital record specifications 110
  using 73
vital record specification
  ABEND 59
  chain 101
  cleaning up 103
  COUNT(0) 103
  data set
    adding 69, 236
    changing 70
    deleting 318
    details 129, 302, 349
    list 132, 412
    retention 77
  DELETED 60
  examples
    chaining vital record specifications 55
  name
    adding 70, 236
    deleting 71, 318
    details 129, 302, 349
vital record specification *(continued)*
  name *(continued)*
    list 412
  OPEN 61
  volume
    adding 69, 236
    deleting 318
    details 129, 302, 349
    list 132, 412
    retention 77
vital record specification management value
  defining management and retention policy 236
  displaying 337
  matching with data set names 65
  specifying 78, 243
vital record specifications
  adding 477
  adding a data set VRS 477
  adding a location name VRS 478
  adding a retention name VRS 478
  changing 492
  creating a list of 412
  defining 53
  defining retention and movement policies 99
  deleting 318
vital records selection processing
  what you can specify 100
VOL1
  clearing the label information 296
  searching for ISO/ANSI label version information 403
VOL1 label, specifying ISO/ANSI label version 229
volume
  canceling outstanding moves 37
  changing 289
  changing information 269
  changing volume serial number 37
  changing volume type 299
  changing WORM tape status 38
  clearing location information 282
  confirming moves 95
  confirming return to scratch activities 282
  confirming volume movement 36
  date assigned to a user 282
  defining using SCAN output 185
  ejecting 93
  expiration date 46
  handling volume discrepancies 184
  in transit 95
  loan location 94
  marking a stacked volume closed 280
  moving 73, 88
  non-restricted operands or fields 33
  non-scratch
    adding 29
    changing information 31
  not defined to DFSMSrmm 27
  overwriting user volumes 27
  preventing a stacked volume from moving 292
  reclaiming from pending status or scratch status 34
  removing 291
  requesting movement 91
  resetting the ABEND status flag 290
  restricted operands or fields 33

volume *(continued)*
RETAINBY attribute 45, 229, 296
retained by job name 76
retained until expired 74, 75
retained while cataloged 74
retained while open 75
retaining
by generic volume serial number 69
by specific volume serial number 69
retention date 82
retention method 45
understanding volume capacity 21
understanding volume compression 21
understanding volume usage 21
updating information 269
volume catalog
obtaining information about volumes 235
VOLUME fast path command 513
volume label
scanning a tape volume 184
VOLUME operand 2
Volume Search panel
using 144
volume serial number
definition 69
generic volume serial number 69
volume usage count 348
volume usage value 348
volume vital record specification
adding 69
definition 53
volumes
adding volume information 211
changing volume information 33
deleting information about 314
deleting volume information 34
displaying information about 344
non-SCRATCH volume optional operands 475
releasing 113
requesting 113
requesting return after release 19
retaining 19
rules for changing volume information 32
setting release actions for 18
tasks associated with 16
using 16
volume operands 470
VRS fast path command 513
VRS operand 2
VRSBYJOBNAME
defining a vital record specification 76
defining job names 58
filtering with job name masks 62
VRSEL managed
source data set
retention of 125
VRSEL processing
excluding data sets from 255, 263, 368
VRSELEXCLUDE 255, 263, 368
VRSEL retention method
compared to EXPDT retention method 48
introduction 45
using 20
VRSELEXCLUDE 255, 263, 368

W

waiting requests, number of 155
while cataloged
catalog date format 83
data sets retained while cataloged 74
minimum retention with WHILECATALOG 75, 249
with special dates 249
while cataloged retention type 74
WHILECATALOG
retention type 74
WORM tape
adding WORM tapes 211
changing WORM tape status 38
identifying WORM tapes ready for destruction 37
identifying WORM tapes ready for reuse 37
wrong data set name, changing 259
wrong label processing 178
wrong volume serial number
correcting 289

Product Number: 5650-ZOS

SC23-6873-50