IBM

Converting to DFSMSrmm from TLMS

- Learn about the process and tools for converting to DFSMSrmm
- Prepare a successful conversion plan
- Run DFSMSrmm in paralel while converting

Archived

Mary Lovelace
Norbert Schlumberger

Redbooks

ibm.com/redbooks

Archived

International Technical Support Organization

Converting to DFSMSrmm from TLMS

February 2004

Archived

SG24-6242-01

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

Archived

Second Edition (February 2004)

This edition applies to Version 1, Release 5, of z/OS DFSMSrmm (product number 5647-A01).

© Copyright International Business Machines Corporation 2001 2004. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures xiii

Tables xxi

Examples xxiii

Notices xxxi
Trademarks xxxii

Preface xxxiii
The team that wrote this redbook xxxiii
Become a published author xxxiv
Comments welcome xxxv

Chapter 1. Introduction 1
1.1 DFSMSrmm overview 2
1.1.1 Environment 2
1.1.2 How is DFSMSrmm structured? 7
1.1.3 What are the DFSMSrmm interfaces? 8
1.1.4 Library and storage location management 15
1.1.5 Media types 21
1.1.6 Volume management 23
1.1.7 Policies for retention and movement 31
1.1.8 DFSMSrmm automated tape library support 37
1.1.9 Catalog synchronization 41
1.2 Conversion process overview 42

Chapter 2. Planning for conversion 47
2.1 Planning considerations 48
2.1.1 Software maintenance levels 48
2.1.2 Timing of events in your installation 48
2.1.3 Change management 48
2.1.4 Limiting business risk 49
2.2 Conversion process 49
2.3 Conversion stages 51
2.3.1 Preparing a conversion plan 52
2.3.2 Learning about DFSMSrmm 54
2.3.3 Starting DFSMSrmm 54
2.3.4 Analyzing the current environment 55

© Copyright IBM Corp. 2004. All rights reserved.

2.3.5 Extracting data for conversion 55
2.3.6 Completing conversion 56
2.3.7 Parallel running and validation 56
2.3.8 Cutover to production 56
2.3.9 Measuring success 57
2.3.10 Full exploitation 57
2.3.11 Uninstall all vendor tape management code from your system . . . 57

Chapter 3. Preparing the environment 59
3.1 DFSMSrmm implementation 60
3.1.1 Installation wizards 60
3.1.2 Update installation-wide exits 61
3.1.3 Update and validate PARMLIB members 62
3.1.4 Creating a starting procedure 85
3.1.5 Define an alias for high-level qualifier RMM 88
3.1.6 Protecting DFSMSrmm resources 89
3.1.7 Defining RACF groups for DFSMSrmm 90
3.1.8 Create the DFSMSrmm control data set (CDS) 91
3.1.9 Create the DFSMSrmm CDS as extended format 95
3.1.10 Create the DFSMSrmm journal 97
3.1.11 Restart MVS with DFSMSrmm implemented 100
3.1.12 Starting and restarting DFSMSrmm 101
3.1.13 Stopping DFSMSrmm 102
3.1.14 Setting up DFSMSrmm utilities 104
3.1.15 Running the installation verification program 105
3.2 Tailoring the DFSMSrmm ISPF dialog 106
3.2.1 Adding DFSMSrmm to ISPF panels 106
3.2.2 Modifying the DFSMSrmm panel 107
3.2.3 Other changes 109
3.3 Making TSO HELP information available to users 109
3.4 Verifying your DFSMSrmm implementation 110
3.4.1 Display PARMLIB options and control information 110
3.4.2 Add owner information 113
3.4.3 Add racks to DFSMSrmm 115
3.4.4 Add volumes to DFSMSrmm 116
3.4.5 Add bins to DFSMSrmm 118
3.4.6 Add a VRS to DFSMSrmm 119
3.4.7 Create tape data sets 120
3.4.8 Allocate inventory management data sets 122
3.4.9 Run inventory management 123
3.4.10 Confirm movements 124
3.4.11 Create reports 125
3.4.12 Run EDGINERS to initialize volumes automatically 126

3.4.13 Restore the CDS . . . 127
3.4.14 Verify the CDS . . . 131
3.4.15 Testing implemented exits . . . 132
3.4.16 Testing application use of tape . . . 135
3.5 Education . . . 136
3.6 New operator procedures . . . 136

Chapter 4. DFSMSrmm ACS support . . . 137
4.1 DFSMSrmm SMS ACS support . . . 138
4.1.1 How it works . . . 140
4.1.2 ACS support for non-system managed volumes . . . 140
4.1.3 Non-system managed tape libraries . . . 143
4.1.4 SMS read-only variables . . . 143
4.1.5 Implementing SMS ACS processing . . . 149
4.1.6 Migration considerations . . . 165
4.2 Partitioning a SMS library . . . 169
4.3 Using DFSMSrmm with virtual tape solutions . . . 174
4.3.1 VTS support for stacked volumes . . . 175
4.3.2 Stacked volume support for non-VTS . . . 179

Chapter 5. TLMS and DFSMSrmm . . . 183
5.1 Terminology . . . 184
5.2 Functions . . . 185
5.3 Retention methods . . . 188
5.4 Data fields . . . 190
5.4.1 Volume . . . 190
5.4.2 Data set . . . 192
5.4.3 Data unavailable for conversion . . . 192
5.5 Cross-reference . . . 193
5.5.1 Volume information . . . 193
5.5.2 Data set name information . . . 198
5.5.3 Vital record information . . . 199
5.5.4 Empty rack and bin information . . . 200
5.5.5 Owner record information . . . 201
5.6 Differences . . . 201
5.6.1 Structure . . . 202
5.6.2 Open, close, and EOV interface . . . 202
5.6.3 Shelf management . . . 202
5.6.4 Retention . . . 203
5.6.5 Deletion policies . . . 203

Chapter 6. Data extraction . . . 205
6.1 Samples and programs . . . 206
6.1.1 Sample data extraction jobs provided with DFSMSrmm . . . 206

Archived

6.1.2 Sample extraction REXX execs provided with DFSMSrmm 206
6.1.3 Conversion programs and exits provided with DFSMSrmm 207
6.1.4 Macros provided with DFSMSrmm 207
6.1.5 Sample exits described in this book 207
6.2 Reviewing the current environment 208
6.2.1 Collecting information 208
6.2.2 Analyzing the record layouts 209
6.2.3 TLMS PARMLIB options 209
6.2.4 Inventory management and reports 212
6.2.5 Exits 214
6.2.6 Data set overwrite 215
6.2.7 BLP and NL processing 216
6.2.8 DFSMShsm interface 217
6.2.9 Tape initialization 218
6.2.10 Tape pool management 219
6.2.11 Security 220
6.3 Retention and movement 221
6.3.1 Retention management 221
6.3.2 Retention criteria 229
6.3.3 Storage location management 232
6.3.4 Movement policies 238
6.4 Interfaces 240
6.4.1 SMS interface 241
6.4.2 BTLS interface 241
6.4.3 Short-on-scratch processing interface 241
6.4.4 TSO and ISPF interfaces 241
6.4.5 Other interfaces 242
6.5 Sample extraction programs 242
6.6 TLMS data extraction 243
6.6.1 Preparing to run the extract program 243
6.6.2 The EDGCDYNM program 243
6.6.3 EDGCDYNM program limitations 255
6.6.4 EDGCDYNM program return codes 255
6.6.5 EDGCDYNM program messages 255
6.7 Am I ready to continue? 260

Chapter 7. Building the CDS 263
7.1 Preparing to run EDGCNVT 264
7.1.1 System-managed volumes 264
7.1.2 Pool prefixes 265
7.1.3 VRS conversion 265
7.1.4 Movement tracking date 266
7.2 EDGCNVT and EDGCEBIN processing 266

Archived

7.2.1 Running EDGCNVT and EDGCEBIN 267
7.2.2 EDGCNVT execution parameter 269
7.2.3 EDGCNVT input files 269
7.2.4 EDGCNVT output files 284
7.2.5 EDGCNVT return codes 286
7.2.6 EDGCNVT messages 286
7.3 EDGCEBIN processing 295
7.3.1 EDGCEBIN execution parameter 296
7.3.2 EDGCEBIN input files 296
7.3.3 EDGCEBIN output files 296
7.3.4 EDGCEBIN return codes 297
7.3.5 6.3.5 EDGCEBIN messages 297
7.4 EDGRACS processing 298
7.4.1 EDGRACS input files 299
7.4.2 EDGRACS output files 299
7.4.3 EDGRACS return codes 299
7.5 EDGCNXIT user exit 300
7.5.1 EDGCNXIT input 301
7.5.2 EDGCNXIT output 301
7.6 DFSMSrmm CDS creation 302
7.6.1 Run EDGJLOAD 302
7.6.2 Create the DFSMSrmm CDS control record 304
7.7 Verifying the DFSMSrmm CDS 305
7.8 Running DFSMSrmm commands 306
7.8.1 Adding empty bins 306
7.8.2 Adding VRSs 307

Chapter 8. Parallel running and validation 309
8.1 Preparing to run in parallel 310
8.1.1 Validate PARMLIB members 310
8.1.2 IEFSSNxx - Subsystem definition 310
8.1.3 EDGRMMxx - DFSMSrmm options 311
8.1.4 Check the DFSMSrmm procedure 321
8.1.5 Exits 321
8.1.6 CA-Disk interface 321
8.1.7 Verify authorization and security 322
8.1.8 Prepare inventory management using EDGHSKP 323
8.1.9 Inventory and movement reporting using EDGRPTD 329
8.1.10 Catalog synchronization 329
8.1.11 Check interfaces 342
8.2 Parallel processing 344
8.3 Starting tape and inventory management 345
8.3.1 First run of EDGHSKP with PARM(VRSEL) 346

Archived

8.3.2 Disabling the journal data set . . . 347
8.3.3 Enabling the journal data set . . . 349
8.4 Validating DFSMSrmm operation . . . 349
8.4.1 Data set name list of non-scratch volumes . . . 349
8.4.2 Scratch volume lists . . . 350
8.4.3 Inventory by location . . . 350
8.4.4 Multifile multi-volume reports . . . 351
8.4.5 Comparing results . . . 351
8.4.6 Checking WTORs and messages . . . 354
8.5 Keeping databases synchronized . . . 354

Chapter 9. Cutover to production . . . 357
9.1 EDGRMMxx PARMLIB options . . . 358
9.1.1 OPTION command . . . 361
9.1.2 VLPOOL command . . . 364
9.2 Control access to DFSMSrmm functions . . . 365
9.3 Activate exits . . . 365
9.4 Remove TLMS . . . 366
9.5 Modifying CA-Disk . . . 366
9.6 Inventory management . . . 367
9.7 Interface to tape libraries . . . 368
9.7.1 System-managed IBM 3494 and IBM 3495 tape libraries . . . 368
9.7.2 Other tape libraries . . . 370
9.8 Starting DFSMSrmm . . . 375
9.9 Final testing . . . 377
9.9.1 DB2 logging and backup . . . 377
9.9.2 RACF tape protection . . . 378
9.9.3 Robot interface . . . 378
9.9.4 Non-system managed libraries SMS ACS support . . . 379
9.10 Commitment to DFSMSrmm . . . 380
9.11 Final activities . . . 383
9.11.1 Uninstalling TLMS . . . 383
9.11.2 Cleaning up your DFSMSrmm implementation . . . 383
9.11.3 Implementing additional functions . . . 388

Chapter 10. Policy management . . . 391
10.1 VRS related PARMLIB options . . . 392
10.1.1 CATRETPD . . . 392
10.1.2 MOVEBY . . . 393
10.1.3 RETAINBY . . . 394
10.1.4 REUSEBIN . . . 396
10.1.5 VRSCHANGE . . . 397
10.1.6 VRSJOBNAME . . . 398

10.1.7 VRSMIN 399
10.1.8 VRSEL 399
10.2 VRS types 401
10.2.1 Data set VRS 402
10.2.2 Volume VRS 408
10.2.3 Name VRS 409
10.3 Specifying VRS parameters 409
10.3.1 DSNAME and NAME retention types 410
10.3.2 Special NAME retention type 412
10.3.3 Volume retention types 413
10.3.4 Retention limits 413
10.3.5 Movement policies 414
10.3.6 Release options 415
10.4 Chaining VRSs 415
10.4.1 VRS chain and subchain 417
10.5 Managing DFSMShsm tapes 418
10.6 DFSMS ACS support 419
10.7 Modifying VRS 420
10.8 Deleting VRS 420
10.9 Trial run 421
10.10 VRS examples 422
10.10.1 ADDVRS operand defaults 422
10.10.2 Adding a single VRS 422
10.10.3 VRS chain with one NEXTVRS operand 423
10.10.4 VRS chain with two ANDVRS operands 427
10.10.5 VRS chain with two subchains 428
10.11 Complex VRS chain with three subchains 429
10.12 VRS hints and tips 430

Chapter 11. Tailoring your DFSMSrmm environment 433
11.1 IBM Tivoli Workload Scheduler 434
11.1.1 IBM Tivoli Workload Scheduler overview 434
11.1.2 Schedule your inventory management tasks 435
11.1.3 Preparation 437
11.1.4 CDS verify 438
11.1.5 CDS backup 438
11.1.6 Inventory management 439
11.1.7 Erasing and labeling volumes 441
11.1.8 Scratch processing 441
11.1.9 Scratch reporting 442
11.1.10 Ejecting volumes from a system-managed tape library 442
11.1.11 Producing reports 443
11.1.12 Move confirmation 443

Archived

11.1.13 Additional considerations 444
11.2 Activate RACF TAPEVOL and TAPEDSN 445
11.2.1 Before you can start 446
11.2.2 DFSMSrmm RACF tape security support 447
11.2.3 DFSMSrmm automatic tape security support 447
11.2.4 DFSMSrmm OPTION command operand TPRACF 447
11.2.5 DFSMSrmm VLPOOL command operand RACF 448
11.2.6 Running DFSMShsm and DFSMSrmm. 449
11.2.7 Define RACF profiles 449
11.2.8 Final RACF activities. 450
11.3 Maintaining your control data set (CDS) 451
11.3.1 Using EDGHSKP 451
11.3.2 Using EDGBKUP 455
11.3.3 EDGUTIL. 461
11.4 Initialize and erase volumes 463
11.5 Disposition processing 464
11.6 Software products and foreign tapes. 466
11.6.1 Defining software products 466
11.6.2 Foreign tapes 467
11.7 DFSMSrmm volume pools 468
11.7.1 Defining pools 469
11.7.2 Scratch pooling 471
11.8 Delete non-existent volumes 472
11.9 Repairing actions. 473
11.9.1 Verify 474
11.9.2 Mend 475
11.9.3 DFSMSrmm CDS and DFSMShsm consistency checking 478
11.9.4 Using DFSMSrmm with BTLS 479
11.9.5 Using DFSMSrmm with non-IBM libraries. 481

Chapter 12. Reporting on your DFSMSrmm environment 485
12.1 Requesting DFSMSrmm resource information 486
12.1.1 Working with resource lists 486
12.1.2 Displaying information from DFSMSrmm 505
12.2 Reporting utilities. 525
12.2.1 Security and audit reports 525
12.2.2 Inventory management reports 526
12.2.3 Inventory, movement and scratch list reports 527
12.2.4 DFSMSrmm-supplied EXECs 529
12.3 DFSORT reporting 539
12.4 DFSMSrmm report generator 540
12.4.1 Setting up the report generator 541
12.4.2 The report type support. 544

12.4.3 The reporting tool support . . . 545
12.4.4 The report definition support . . . 547
12.4.5 Using the report generator . . . 547
12.4.6 Report generator messages . . . 558
12.5 Sample reports . . . 558

Chapter 13. Splitting and merging your DFSMSrmm CDS . . . 561
13.1 When to merge or split . . . 562
13.1.1 Considerations . . . 562
13.1.2 PARMLIB options . . . 570
13.1.3 Merging the CDS . . . 572
13.1.4 Splitting the CDS . . . 576
13.1.5 Converting to an existing CDS . . . 583
13.2 Merging DFSMSrmm environments . . . 583
13.2.1 One or more RMMplexes . . . 583
13.2.2 Considerations for merging RMMplexes . . . 584
13.2.3 Methodology for merging RMMplexes . . . 598
13.2.4 Tools and documentation . . . 602

Appendix A. Conversion checklist . . . 607

Appendix B. Conversion record layouts . . . 613
Record layout description . . . 614
EDGCDREC record layout . . . 615
Constants . . . 618
Cross reference . . . 618
EDGCEREC record layout . . . 620
Constants . . . 621
Cross reference . . . 621
EDGCKREC record layout . . . 623
Constants . . . 625
Cross reference . . . 625
EDGCLREC record layout . . . 628
Constants . . . 635
Cross reference . . . 635
EDGCOREC record layout . . . 643
Constants . . . 644
Cross reference . . . 644

Appendix C. Security topics . . . 647
C.1 RACF implementation . . . 648
C.1.1 Assign a RACF user ID for DFSMSrmm . . . 648
C.1.2 Identify DFSMSrmm to RACF . . . 649
C.1.3 Define DFSMSrmm resources to RACF . . . 650

Archived

C.1.4 Define RACF groups for DFSMSrmm users 652
C.2 CA-Top Secret implementation 670
C.2.1 Define user groups 670
C.3 ACF2 implementation 674
C.3.1 Define data set rules 674
C.3.2 Define DFSMSrmm resources 675

Appendix D. Additional material 679
Locating the Web material 679
Using the Web material 679
System requirements for downloading the Web material 680
How to use the Web material 680

Glossary 681

Related publications 697
IBM Redbooks 697
Other resources 697
Referenced Web sites 697
How to get IBM Redbooks 698
IBM Redbooks collections 698

Index 699

Archive

Figures

1-1 DFSMSrmm environment 2
1-2 List of volumes owned by a single user 5
1-3 DFSMSrmm structure 7
1-4 DFSMSrmm in a multisystem environment 8
1-5 Interfaces between DFSMSrmm and other software components 9
1-6 SMS tape allocation 11
1-7 High Speed Locate overview 13
1-8 DFSMSrmm API 14
1-9 Removable media libraries and storage locations 16
1-10 Internal labels, external labels, and rack numbers 17
1-11 System managed tape overview 18
1-12 Support of different home locations 21
1-13 Different cartridge media types 22
1-14 Tape drive matrix 22
1-15 Stacked, logical, and physical volumes 23
1-16 Life cycle of a volume 25
1-17 Timing of related events for a volume 26
1-18 Global ignore volume RACF resource 27
1-19 RACF ignore volume resources 27
1-20 Bypass DFSMSrmm function 28
1-21 DFSMSrmm duplicate volume support 29
1-22 Flow chart to add a duplicate volume serial number 30
1-23 Add duplicate volume 31
1-24 RETAINBY(SET): Expiration date 34
1-25 DFSMSrmm and SMS ACS support 35
1-26 Automated tape library (ATL) environment 38
1-27 REJECT command examples 39
1-28 Catalog status tracking 42
1-29 Conversion process overview 43
2-1 Conversion process flow 51
2-2 Sample schedule 53
3-1 Defining DFSMSrmm subsystem name 63
3-2 Defining DFSMSrmm subsystem name with interface enabled 63
3-3 MVS command SETSSI 63
3-4 MVS command SETSSI and activate subsystem interface 64
3-5 MVS command SETSSI 64
3-6 MVS command SETSSI to deactivate subsystem interface 65
3-7 IKJTSOxx operand AUTHCMD 65

© Copyright IBM Corp. 2004. All rights reserved.

3-8 IKJTSOxx operand AUTHTSF 65
3-9 IKJTSOxx operand AUTHPGM 65
3-10 TSO command PARMLIB update 66
3-11 TSO command PARMLIB list. 66
3-12 Enabling DFSMSrmm license into IFAPRDxx 67
3-13 MVS command SET 68
3-14 MVS command DISPLAY 68
3-15 Display SMS information 79
3-16 GRS EXCL definition 81
3-17 GRS CON definition. 81
3-18 Use of GRS 81
3-19 Update GRS information 82
3-20 Display GRS information 82
3-21 MIMQNAME member using hardware reserves 84
3-22 GDIEXMPT member 85
3-23 MIMQNAME member using propagation of ENQs 85
3-24 Display defined system symbols 86
3-25 Add DFSMSrmm user ID to RACF 90
3-26 Assign DFSMSrmm to RACF STARTED class 90
3-27 Data class specifying extended format 96
3-28 EDG0105I message after initialization of the SSI 101
3-29 EDG0103D message to activate the SSI. 101
3-30 Results of F DFRMM,QUERY ACTIVE command. 103
3-31 DFSMSrmm MODIFY command output 103
3-32 S DFRMM,OPT_RESET command output 104
3-33 P DFRMM command output 104
3-34 RMM ISPF selection 106
3-35 ISPF selection with NEWAPPL 106
3-36 ISPF selection without NEWAPPL. 107
3-37 Add DFSMSrmm to a ISPF selection panel. 107
3-38 RMM ISPF selection with NEWAPPL and USER selection 107
3-39 RMM ISPF USER selection using RMMISPF REXX exec. 108
3-40 RMM ISPF selection with NEWAPPL and LIBRARIAN selection 108
3-41 RMM ISPF LIBRARIAN selection using RMMISPF REXX exec 108
3-42 Confirm move for a single volume 124
3-43 Confirm move for all volumes 124
3-44 QUISCE all DFSMSrmm activities. 127
3-45 Restart DFSMSrmm. 128
3-46 CDS recovering from latest backup 128
3-47 CDS recovering from backup-1 129
3-48 RMM ADDVOLUME subcommand with location ATL 133
3-49 Display OAM status 134
4-1 SMS ACS support with DFSMSrmm V2R10 or higher 138

4-2 DFSMSrmm ACS processing PARMLIB options 139
4-3 How DFSMSrmm SMS direct calls for non-system managed volumes 141
4-4 DFSMSrmm EDGUX100 user exit call with PREACS(YES) 142
4-5 DFSMSrmm SMS direct and EDGUX100 user exit call 142
4-6 ISMF primary option menu 150
4-7 ISMF library selection menu 151
4-8 Specify a library name to SMS for a non-system managed library . . . 151
4-9 Define a new SMS tape library 1 of 2 152
4-10 Define a new SMS tape library 2 of 2 152
4-11 Restart OAM to activate the new tape library 153
4-12 Use MVS display SMS command 153
4-13 Specify a storage group name to SMS 154
4-14 Define a new SMS storage group 155
4-15 SMS STORAGE GROUP STATUS DEFINE 156
4-16 List all defined SMS storage groups 156
4-17 Restart SMS to activate the new tape library 157
4-18 ISMF management class selection 158
4-19 ISMF Panel: Management Class Define 158
4-20 Restart SMS to activate the new tape library 159
4-21 Restart DFSMSrmm after updating EDGRMMxx 163
4-22 Add 10 new scratch volume to pool L* 164
4-23 Update volumes storage group information 164
4-24 Define new data set name vital record specifications 165
4-25 DFSMSrmm V2R10 SMS ACS VRS policy management support 167
4-26 Volumes enter a tape library 170
4-27 Scenario with one CDS and one TCDB 171
4-28 Scenario with two CDSs and one TCDB 172
4-29 Scenario with two CDSs and two TCDBs 173
4-30 Scenario with two TCDBs and one CDS 173
4-31 DFSMSrmm Control Record Display panel 175
4-32 EDGUTIL example 175
4-33 VTS support for stacked volumes 176
4-34 VRSEL processing output 177
4-35 Export process for non-VTS 180
4-36 Import process for non-VTS 180
6-1 VRS hierarchy 224
6-2 Define VRS for data set name retention 227
6-3 Define VRS for retention while cataloged 228
6-4 TLMSLOCN statements 229
6-5 VRS data set mask conversion 231
6-6 Define VRS for data set name movement 233
6-7 Define VRS for application movement policy 234
6-8 Define VRS for GDG data set name 235

Archived

6-9 Define VRS for application first store 236
6-10 Define name VRS 236
6-11 Define VRS for application second store 237
6-12 Cabinet expanding until conversion 240
6-13 Box slot number expanding until conversion 240
6-14 EDGCDYNM SYSIN: OPTIONS control statement 245
6-15 EDGCDYNM SYSIN: VRSNAMEP control statement 247
6-16 EDGCDYNM SYSIN: VRSMGMTP control statement 248
6-17 DFSMSrmm VRS management value 248
6-18 EDGCDYNM SYSIN: TLMSLOCN control statement 249
6-19 TLMSLOCN mapping. 249
6-20 EDGCDYNM SYSIN: DFRMMLOC control statement 250
6-21 EDGCDYNM DFRMMLOC mapping 250
6-22 EDGCDYNM SYSIN: VOLMEDIA control statement 251
6-23 SKIPVOL control statement 252
6-24 EDGCDYNM SYSIN: DEFOWNER control statement. 252
6-25 EDGCDYNM SYSIN: VRSOWNER control statement. 253
7-1 Convert program VRS input record format 271
7-2 EDGCNVT record create date, time, and system identification 271
7-3 VRSEL release action 271
7-4 RETACTN release action. 272
7-5 IF STORLOC SYSIN statement. 273
7-6 IF STORLOC example with LOCDEF 273
7-7 IF STORLOC onsite example 273
7-8 IF STORLOC offsite example 274
7-9 IF STORLOC remote example. 274
7-10 IF STORLOC with blank example 274
7-11 IF DSN SYSIN statement syntax 274
7-12 IF VOL SYSIN statement syntax 275
7-13 IF OWNER SYSIN statement syntax for default owner 275
7-14 IF OWNER SYSIN statement syntax for translating owner 276
7-15 LOCDEF statement 276
7-16 VOLPRE SYSIN statement 278
7-17 VOLPRE example 278
7-18 VOLRANGE with LOCATION statement 278
7-19 VOLRANGE with PREFIX statement. 279
7-20 VOLRANGE with PREFIX statement. 280
7-21 VOLRANGE with MEDIANAME statement 280
7-22 VOLRANGE with SGNAME statement 280
7-23 VOLRANGE with TYPE statement 281
7-24 VOLRANGE with NORACK statement 281
7-25 VOLRANGE with VLPOOL statement 282
7-26 VMEDIA statement with VLPOOLNAME 282

7-27 VMEDIA with MEDIANAME statement 283
7-28 Example of SYSOUT contents. 284
7-29 Example of VRSCMDS contents 285
7-30 Message IDC3314I 304
8-1 Defining DFSMSrmm subsystem name. 311
8-2 Disabling the DFSMSrmm subsystem interface 322
8-3 Catalog status tracking 331
8-4 List of all used system IDs 332
8-5 TSO RMM subcommands to update data set records. 335
8-6 Enabling catalog synchronization 337
9-1 EDG4027I message. 377
9-2 EDG4050I message. 377
9-3 CBR3010I message. 379
10-1 How catalog retention works 393
10-2 RETAINBY(SET) MOVEBY(SET) 394
10-3 Retain by set indicator 394
10-4 Volume set. 395
10-5 RETAINBY(SET) - EXPDT 395
10-6 RETAINBY(SET) - VRS. 396
10-7 RESUSE bin with CONFIRMMOVE operand 396
10-8 RESUSE bin with STARTMOVE operand 397
10-9 VRS examples using wild card characters. 398
10-10 Syntax string using VRSJOBNAME(2) 398
10-11 Syntax string using VRSJOBNAME(2) 399
10-12 Define two data set name VRS definitions. 400
10-13 Data set created using a management value of CYCL004 400
10-14 DFSMSrmm Add Vital Record Specification panel 402
10-15 Selecting matching VRS for data sets (new) 403
10-16 VRSEL(OLD) best match fit sequence 404
10-17 Data set name matching order. 408
10-18 VRS chain with three subchains 416
10-19 VRS chain with two subchains. 416
10-20 VRS chain 417
10-21 VRS chain with two subchains. 417
10-22 Support before OS/390 V2R10 for non-system-managed tapes 419
10-23 DFSMSrmm VRSs panel 420
10-24 DFSMSrmm Data Sets panel. 421
10-25 Inventory management trial run 421
10-26 Single VRS example 423
10-27 Single VRS Retention and location results 423
10-28 VRS chain with one NEXTVRS operand 424
10-29 Single VRS using the NEXTVRS operand results 424
10-30 VRS chain with Two NEXTVRS operands. 425

10-31 Single VRS using two NEXTVRS operand results 425
10-32 VRS chain with retention information example 426
10-33 Single VRS using two retention NEXTVRS operands results 427
10-34 VRS chain with two ANDVRS operands example 427
10-35 Single VRS using two retention NEXTVRS operands results 428
10-36 VRS chain with Two Subchains Example 428
10-37 ANDVRS and NEXTVRS with retention operand results 429
10-38 VRS chain with three subchains example 430
10-39 Data set examples 432
11-1 DFSMSrmm IBM Tivoli Workload Scheduler for z/OS job flow 437
11-2 PARMLIB setting for automatic backup 454
11-3 Default label format for a tape cartridge 464
11-4 EDG5054I message display 465
11-5 Example label format 465
11-6 Sample pools definitions 471
11-7 Three-way audit of CDS, TCDB and LM DB 474
11-8 EDGUTIL MEND syntax diagram 474
11-9 EDGUTIL MEND syntax diagram 476
11-10 DFSMSrmm CDS,Volume Catalog,Library Manager constancy check 477
11-11 EDGUTIL MEND(SMSTAPE) does not update the CDS 478
12-1 Reaching the Rack and Bin Search panel 487
12-2 Searching for shelf locations 488
12-3 List of SCRATCH shelf locations 489
12-4 RMM TSO SEARCHRACK command output 489
12-5 Reaching the Volume Search panel 490
12-6 Searching for volumes 491
12-7 Volumes list (1 of 2) 492
12-8 Volumes list (2 of 2) 492
12-9 RMM SEARCHVOLUME command output 493
12-10 Reaching the Product Search panel 493
12-11 Product Search panel 494
12-12 Product list (1 of 2) 494
12-13 Product list (2 of 2) 495
12-14 RMM SEARCHPRODUCT command output 495
12-15 Reaching the Data Set Search panel 495
12-16 Data Set Search panel 496
12-17 Data set list 497
12-18 RMM SEARCHDATASET command output 497
12-19 Reaching the Search VRSs panel 498
12-20 Search VRSs panel 498
12-21 VRS list (1 of 3) 499
12-22 VRS list (2 of 3) 499
12-23 VRS list (3 of 3) 500

12-24 SEARCHVRS command output 500
12-25 Dialog Sort Options Menu panel 501
12-26 Volume List Sort Options panel 501
12-27 Volume list with default sorting options 502
12-28 Volume list with changed sorting options 502
12-29 Volume list using SORT command 503
12-30 Reaching Volume Display panel 506
12-31 Removable Media Manager primary menu 506
12-32 DFSMSrmm Command Menu 507
12-33 DFSMSrmm Volume Menu 507
12-34 DFSMSrmm Volume Display 508
12-35 DFSMSrmm Volume Details menu (1 of 2) 508
12-36 DFSMSrmm Volume Details menu (1 of 2) 509
12-37 RMM TSO LISTVOLUME command output (1 of 2) 510
12-38 RMM TSO LISTVOLUME command output (2 of 2) 511
12-39 Removable Media Manager primary panel 511
12-40 DFSMSrmm Command Menu 512
12-41 DFSMSrmm Data Set Menu 512
12-42 DFSMSrmm Data Set Display panel 513
12-43 DFSMSrmm Data Set Details panel (1 of 2) 513
12-44 DFSMSrmm Data Set Details panel (2 of 2) 514
12-45 RMM TSO LISTDATASET command output 514
12-46 Reaching the Rack and Bin Display panel 515
12-47 DFSMSrmm primary menu 515
12-48 DFSMSrmm Command Menu 516
12-49 DFSMSrmm Rack and Bin Menu 516
12-50 DFSMSrmm Rack and Bin Display 517
12-51 DFSMSrmm Rack or Bin Details 517
12-52 RMM TSO LISTRACK command output 518
12-53 Reaching the Display VRSs panel 518
12-54 DFSMSrmm Primary menu 518
12-55 DFSMSrmm Command Menu 519
12-56 DFSMSrmm Vital Record Specification Menu 519
12-57 Display Virtual Record Specification 520
12-58 DFSMSrmm Display Data Set VRS panel 520
12-59 RMM TSO LISTVRS command output 521
12-60 DFSMSrmm Primary menu 521
12-61 DFSMSrmm Command Menu panel 522
12-62 DFSMSrmm Control Information Menu 522
12-63 DFSMSrmm System Options Display panel 523
12-64 RMM TSO LISTCONTROL command output 524
12-65 JCL to generate security and audit reports 525
12-66 Sample JCL for inventory., movement, and scratch list reporting 527

12-67 Report selection 532
12-68 Creating a report security header 532
12-69 Defining a CCARD 533
12-70 Sorting by volume serial number and volume status 534
12-71 Sorting by volume serial number, status, and temporary errors 534
12-72 REPORT01 report header 534
12-73 REPORT01 report header modified 535
12-74 REPORT01 column headings 535
12-75 REPORT01 column headings modified 535
12-76 REPORT01 returned values 536
12-77 REPORT01 returned values modified 536
12-78 Sample REPORT01 output with new header 537
12-79 Sample REPORT11 output with new Vol-Cnt. column 538
12-80 Dialog Options Menu panel 542
12-81 DFSMSrmm Report Options panel 543
12-82 Command Menu panel with REPORT option 547
12-83 User Menu panel with REPORT option 548
12-84 Librarian Menu panel with REPORT option 548
12-85 Report Generator panel 549
12-86 Report Definition Search panel 549
12-87 Report Definitions panel 550
12-88 New report name prompt 550
12-89 Select Report Type panel 551
12-90 Select Reporting Tool panel 551
12-91 Report Definition panel with criteria fields 552
12-92 Report Criteria panel 553
12-93 Report Criteria Details panel 553
12-94 Reporting data set name prompt 554
12-95 Delete report definition confirmation prompt 554
12-96 Report Types panel 555
12-97 Report Type panel 556
12-98 Report Type Criteria panel 557
12-99 Reporting Tools panel 557
13-1 Merging process for multiple RMM CDSs 586
C-1 RACF command ADDUSER 648
C-2 Add DFRMM to RACF started class 649
C-3 Add the concurrent copy resource to RACF 663
C-4 HOUSEKEEP resource for EDGBKUP 667
C-5 RMM data set profiles for EDGBKUP 667
C-6 HOUSEKEEP resource for EDGXPROC 668
C-7 RMM data set profiles for EDGXPROC 668
C-8 OPERATOR resource for EDGLABEL 669

Tables

1-1 DFSMSrmm ignore profile checking 28
3-1 DFSMSrmm CDS space requirement 92
3-2 Space requirement for journal data set record. 97
3-3 DFSMSrmm libraries and their default DD names. 108
4-1 SMS read-only variables set by DFSMSrmm 143
4-2 Stacked volume information. 178
5-1 TLMS and DFSMSrmm terminology comparison 184
5-2 TLMS and DFSMSrmm functions comparison. 185
5-3 TLMS retention methods and DFSMSrmm VRSs equivalences 188
5-4 TLMS and DFSMSrmm volume data fields 190
5-5 TLMS and DFSMSrmm data set data fields 192
5-6 EDGCLREC volume information 193
5-7 EDGCDREC data set name information 198
5-8 EDGCKREC policy information 199
5-9 EDGCEREC empty rack and bin information 200
5-10 EDGCOREC owner information. 201
6-1 DFSMSrmm provided sample jobs 206
6-2 DFSMSrmm-supplied conversion REXX execs. 206
6-3 DFSMSrmm-supplied conversion programs and exits. 207
6-4 DFSMSrmm-supplied macros 207
6-5 DFSMSrmm provided sample exits 207
6-6 TLMS data set name types 221
6-7 IBM-provided extraction samples for TLMS. 242
6-8 EDGCDYNM output data sets and records 244
7-1 EDGCNVT execution parameters 269
7-2 EDGCNVT input files 269
7-3 EDGCNVT output files. 284
7-4 EDGCNVT return codes 286
7-5 EDGCEBIN execution parameters. 296
7-6 EDGCEBIN input files 296
7-7 EDGCEBIN output files 296
7-8 EDGCEBIN return codes 297
7-9 EDGRACS input files. 299
7-10 EDGRACS output files. 299
7-11 EDGCEBIN return codes 299
7-12 EDGCNXIT return codes 301
8-1 Date formats 315
8-2 EDGHSKP input files 324

© Copyright IBM Corp. 2004. All rights reserved.

8-3 EDGHSKP output files 324
8-4 EDGHSKP execution parameters 327
9-1 Steps required to go into production with DFSMSrmm 376
9-2 Fallback plan 382
10-1 Specifying VRS 409
11-1 Schedule inventory management tasks 436
12-1 Resource list line operators 504
12-2 DFSMSrmm reports 530
12-3 Sample reports 559
13-1 Considerations for merging RMMplexes 584
A-1 Conversion checklist 607
B-1 Record layout description 614
C-1 DFSMSrmm resources and protected functions 650
C-2 Access level for each user group 653

Archived

Examples

1-1 DFSMSrmm TSO command interface 14
1-2 Defining a storage location as home location 20
1-3 RMM ADDVOLUME subcommand example 24
3-1 DISPLAY SSI command output 64
3-2 Output of the TSO PARMLIB list command 66
3-3 Output of a DISPLAY PROD command 68
3-4 Sample EDGRMMxx PARMLIB member 76
3-5 Output of a display SMS command 79
3-6 DISPLAY GRS command output 82
3-7 Display SYMBOLS output 86
3-8 Sample DFSMSrmm starting procedure 87
3-9 Define an alias in an ICF-user catalog 89
3-10 Sample JCL to create the DFSMSrmm CDS 93
3-11 Sample JCL to protect the DFSMSrmm CDS 94
3-12 Sample JCL to initialize the DFSMSrmm CDS 94
3-13 Sample JCL to create an extended format DFSMSrmm CDS 96
3-14 Sample JCL to create the DFSMSrmm journal 98
3-15 Sample JCL to protect the DFSMSrmm CDS 99
3-16 MVS command to start DFSMSrmm 101
3-17 VS command to start DFSMSrmm with parameters 101
3-18 Modify DFSMSrmm using new PARMLIB options 102
3-19 Turn PDA on of off 102
3-20 Quiesce all DFSMSrmm activities 102
3-21 Activate or refresh DFSMSrmm installation exits 102
3-22 Show all outstanding DFSMSrmm requests 102
3-23 Disable the DFSMSrmm SSI 103
3-24 Stopping the DFSMSrmm subsystem 104
3-25 Update the TSO logon procedure 109
3-26 Sample JCL to display PARMLIB options and control information . . . 110
3-27 Sample RMM LISTCONTROL subcommand output 111
3-28 Sample JCL to add owner information 114
3-29 Sample RMM LISTOWNER subcommand output 114
3-30 Sample JCL to add racks 115
3-31 Sample RMM SEARCHRACK subcommand output 115
3-32 Sample JCL to add volume information 116
3-33 Sample RMM SEARCHVOLUME subcommand output 117
3-34 Sample JCL to add bins 118
3-35 Sample RMM SEARCHBIN subcommand output 118

© Copyright IBM Corp. 2004. All rights reserved.

3-36 Sample JCL to add VRS 119
3-37 Sample RMM LISTVRS subcommand output 120
3-38 Sample JCL to create data sets on tape 120
3-39 Sample JCL to pre-allocate EDGHSKP data sets 122
3-40 Sample JCL to run inventory management 123
3-41 Sample JCL for EDGINERS automatic processing 127
3-42 Sample JCL to delete and define the CDS 130
3-43 Sample JCL to forward recover the CDS (backup with AMS) 130
3-44 Sample JCL to forward recover the CDS (backup with DSS) 131
3-45 Sample JCL to verify the CDS contents. 131
4-1 IGD008I SMS activation message 153
4-2 Output of a DISPLAY SMS command 153
4-3 IGD008I SMS activation message 157
4-4 IGD008I SMS activation message 159
4-5 Sample ACS management class routine 159
4-6 Sample ACS management class routine for tape only 161
4-7 ACS storage group routine for non-system managed tape. 162
4-8 EDGRMMxx PARMLIB member to support SMS ACS processing . . . 162
4-9 DFSMSrmm restart messages. 163
4-10 Using REJECT operand. 170
4-11 TSO ADDVOLUME subcommand operand USE. 171
4-12 Example to use the ADDVOLUME subcommand operand USE 171
4-13 Using ADDVOLUME with location parameter 176
4-14 TSO RMM ADDDATASET subcommand 177
4-15 Using TSO RMM SEACHVOLUME subcommand. 177
4-16 Confirm volume movement 177
4-17 Search for volumes should be imported 178
4-18 List a single volume should be imported 178
4-19 Search for logical volumes should be imported 178
4-20 Defining non-VTS stacked volumes. 179
6-1 Sample TLMSOPTS output 210
6-2 RMM add an ABEND VRS. 223
6-3 JCL example to use a special EXPDT 223
6-4 RMM add a VRS for spezial EXPDT processing 223
6-5 Sample retention selection report for migration 230
6-6 Sample converted VRS definitions 230
6-7 EDGCDYNM OPTIONS statement 247
6-8 EDGCDYNM VRSMGMTP statement 248
6-9 Add a VRS definition using TSO subcommands 249
6-10 Sample JCL to execute EDGCDYNM 254
7-1 Sample EDGCNVT execution JCL 267
7-2 Sample EDGCEBIN execution JCL 295
7-3 Sample EDGRACS execution JCL 298

7-4 Sample EDGJLOAD JCL 302
7-5 Sample JCL for CDS create and verify 305
7-6 Sample JCL for CDS verify processing 305
7-7 ADD empty bins 306
8-1 Sample EDGRMMxx PARMLIB member 312
8-2 Sample DFSMSrmm starting procedure 321
8-3 Sample EDGHSKP JCL 323
8-4 Using DSSOPT DD statement 324
8-5 Example of JCL for EDGHSKP 326
8-6 Sample JCL for inventory and movement reporting 329
8-7 Sample JCL to find out all used system IDs 332
8-8 Sample JCL to find out all data sets without create system IDs info . . 333
8-9 Sample JCL to find out all data sets without create system ID info . . . 334
8-10 Sample JCL to execute all TSO RMM CHANGEDATASET command 336
8-11 EDGRMMnn PARMLIB members 336
8-12 JCL to disable catalog synchronization 337
8-13 RMM LISTCONTROL subcommand 338
8-14 Output of the TSO RMM LISTCONTROL subcommand 338
8-15 JCL to allocate a larger DFSMSrmm message data set 339
8-16 JCL for catalog synchronization 339
8-17 Message data set after catalog synchronization 340
8-18 JCL to enable catalog synchronization 341
8-19 RMM LISTCONTROL subcommand 341
8-20 RMM LISTCONTROL output 341
8-21 Add a TAPEVOL profile 343
8-22 Permit HSC to MVCs 343
8-23 Permit HSC to RMM resources 344
8-24 ADD an OPERCMDS resource 344
8-25 PERMIT VSM to OPERCMDS 344
8-26 Starting the DFSMSrmm subsystem 345
8-27 Message EDG0103D 345
8-28 Message EDG2107E 346
8-29 Messages EDG2104E and EDG2103D 347
8-30 Restarting the DFSMSrmm subsystem 347
8-31 Messages EDG0204I, EDG0122I and EDG0105I 347
8-32 LISTCONTROL to list options 348
8-33 LISTCONTROL OPTION output 348
8-34 Restarting the DFSMSrmm subsystem 349
8-35 SEARCHVOLUME to list all volumes in status SCRATCH 350
8-36 SEARCHVOLUME to list a volumes in a specific location 350
9-1 Sample EDGRMMnn PARMLIB member for PROTECT mode 358
9-2 REXX exec to check cartridge ejects 369
9-3 RMM SEARCHVOLUME subcommands with specific destinations . . 370

9-4 Using TSO RMM subcommands in a batch JOB 371
9-5 HSC EJECT statements 371
9-6 Sample EDGINERS JCL 371
9-7 DFSMSrmm and HSC synchronization JCL 372
9-8 Original SLUDRRMM macro 373
9-9 Updating SLUDRRMM macro 373
9-10 Add a TAPEVOL profile 374
9-11 Permit HSC to MVC's 374
9-12 Permit HSC to RMM resources 375
9-13 ADD an OPERCMDS resource 375
9-14 PERMIT VSM to OPERCMDS 375
9-15 RMM LISTVOLUME subcommand 378
9-16 RMM CHANGVOLUME subcommand to change the location 379
9-17 Sample JCL for non-specific tape volume mount 379
9-18 RMM CHANGEVOLUME subcommand to change the volume status 381
9-19 RMM CHANGEVOLUME subcommand to change the retention 381
9-20 RMM CHANGEVOLUME subcommand to confirm the move 381
9-21 RMM CHANGEVOLUME subcommand to move a volume 381
9-22 RMM SEARCHVOLUME with LOCATION subparameter 381
9-23 RMM subcommand ADDVRS DSN(**) 386
9-24 RMM subcommand ADDVRS DSN(*.**) 386
9-25 Modifying DFSMSrmm 386
9-26 RMM CHANGEVOLUME to change the expiration date 387
9-27 Deleting non-existent volumes 388
10-1 Fully qualified VRS definition 405
10-2 Define a VRS with GDG attribute 405
10-3 Define a GDG using wild card characters 406
10-4 Define a VRS with GDG attribute and without a limit 406
10-5 Define a VRS without any limit for a group of data sets 406
10-6 Define a management class VRS 406
10-7 Define a management value VRS 407
10-8 Define an ABEND VRS 407
10-9 Define an OPEN VRS 407
10-10 ADDVRS to retain a specific volume 408
10-11 ADDVRS to retain a group of volumes 408
10-12 ADDVRS chain with on subchain 409
10-13 Define a VRS with CYCLES and GDG 410
10-14 Define a VRS with CYCLES 411
10-15 Define a VRS with DAYS 411
10-16 Define a VRS with LASTREFERENCEDAYS 411
10-17 Define a VRS to retain all data sets until expired 412
10-18 Define a volume VRS for a single volume 413
10-19 Define a volume VRS for a group of volumes 413

10-20 Defining a data set VRS using COUNT and DAYS 413
10-21 Defining a data set VRS using COUNT and CYCLES 414
10-22 ADD a VRS definition with DELETEDATE operand 414
10-23 ADD a VRS to retain DFSMShsm managed data sets 418
10-24 ADD a VRS to expand DFSMShsm release actions 419
10-25 VRS data set name definition with the same retention criteria 431
10-26 VRS management class definition 431
10-27 Define a new management class VRS 431
10-28 ADD a VRS management class definition 431
11-1 EDGHSKP parameter DSTORE 440
11-2 TSO RMM search volume command 442
11-3 TSO RMM change volume command 442
11-4 TSO RMM confirm move command 444
11-5 Activate tape data set protection 445
11-6 RACF commands for TAPEVOL and TAPEDSN implementation 449
11-7 JCL to refresh RACF in-storage-profiles 450
11-8 CDS and journal backing up using AMS 453
11-9 CDS and journal back up using DFSMSdss 453
11-10 Procedure for backing up CDS and journal 454
11-11 CDS and journal backing up using AMS 456
11-12 Quiescing the DFSMSrmm subsystem interface 457
11-13 CDS and journal backing up using AMS 457
11-14 Restart DFSMSrmm after it has been quiesced 458
11-15 Restoring a control data set without forward recovery 458
11-16 Restoring a control data set with forward recovery 459
11-17 Restoring a CDS from older back up with forward recovery 459
11-18 .Recovery using journal data sets backed up by BACKUP(AMS) 460
11-19 Recovery using journal data sets backed up by BACKUP(DSS) 460
11-20 Forward recovery of restored CDS 461
11-21 Set last synchronization date and time 462
11-22 TSO RMM ADDVOLUME subcommand 463
11-23 TSO RMM CHANGEVOLUME subcommand 463
11-24 TSO RMM ADDVOLUME subcommand using VOL1 parameter 463
11-25 Sample JCL to create sticky label 465
11-26 Add a software product VLPOOL 466
11-27 ADD RACK subcommand 466
11-28 ADD a program product 466
11-29 ADD volumes to a product 467
11-30 ADD a VLPOOL to manage foreign volumes 467
11-31 Add 100 new racks 468
11-32 Search for volumes waiting for return release action 468
11-33 Confirm a single volumes release action return 468
11-34 Deleting nonexistent volumes 472

11-35 JCL to verity all information in CDS 475
11-36 JCL to MEND data set and volume records only 476
11-37 Scratch pool definitions 480
11-38 BTLS status synchronization 480
11-39 Search for all volume pending move 482
11-40 Sample EJECT statement example 482
11-41 JCL for initializing cartridges with EDGINERS 483
11-42 DFSMSrmm and HSC synchronization JCL 484
12-1 ample JCL to issue RMM TSO subcommands 487
12-2 TSO search rack 489
12-3 TSO search product 495
12-4 TSO search data set 497
12-5 TSO search vital record specification 500
12-6 TSO list volume 510
12-7 TSO list rack 517
12-8 TSO list DSNAME vital record specification 520
12-9 TSO list CDS control information 524
13-1 Listing product information 566
13-2 Deleting product information 566
13-3 Adding product information 566
13-4 Assigning a new rack number 567
13-5 Copying rack numbers 567
13-6 Change the assigned bin number 568
13-7 Freeing and deleting a duplicate bin number 568
13-8 Sample REPRO statements for copying bin numbers 569
13-9 Sample REPRO statements for copying ranges of volumes 570
13-10 JCL to back up the control data set on SYS2 573
13-11 JCL to Restore the SYS2 control data set on SYS 573
13-12 Merge JCL 574
13-13 Sample JCL to split a DFSMSrmm CDS 578
13-14 Sample JCL to back up the DFSMSrmm CDS and JOUNAL 582
13-15 Create a new owner 590
13-16 Delete owner and move all volume to a new owner 591
13-17 List program product record 591
13-18 Add a new program product 591
13-19 Change the volumes product assignment 591
13-20 Delete old product record 591
13-21 Change the assigned rack number 592
13-22 Change volume to norack and delete the rack 593
13-23 Reassign a bin number 593
13-24 Move a volume to its home location 594
13-25 JCL to back up the DFSMSrmm CDSs and journals 603
13-26 DFSMSrmm merge job 603

13-27 CATSYNCH job 606
C-1 JCL to add DFSMSrmm user ID in RACF STARTED class. 649
C-2 JCL to create RACF groups and connect user IDs 654
C-3 JCL to permit access to DFSMSrmm data set profiles 656
C-4 JCL to Define DFSMSrmm Resources in the FACILOITY CLASS . . . 658
C-5 JCL to define DFSMSdss RACF FACILITY class profile 663
C-6 JCL to define DFSMShsm to DFSMSrmm resources 664
C-7 JCL to define ABARS to DFSMSrmm resources 666
C-8 JCL to define EDGBKUP to DFSMSrmm resources 667
C-9 JCL to define EDGXPROC to DFSMSrmm resources. 668
C-10 JCL to define EDGLABEL to DFSMSrmm resources 670
C-11 CA-Top Secret user groups definitions 670
C-12 ACF2 data set rules definition 674
C-13 ACF2 DFSMSrmm resources definition. 675

Archive

Archived

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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.

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.

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 illustrates 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. 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 IBM's application programming interfaces.

© Copyright IBM Corp. 2004. All rights reserved.

Trademarks

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

CICS®
DB2®
DFSMS/MVS®
DFSMSdfp™
DFSMSdss™
DFSMShsm™
DFSMSrmm™
DFSORT™
IBM®
Magstar®
MVS/ESA™
MVS™
OS/390®
RACF®
Redbooks (logo) ™
Redbooks™
RETAIN®
RMF™
SP™
System/390®
Tivoli®
z/OS™

Archived

Preface

DFSMSrmm™ is the IBM® tape management system for OS/390® and z/OS™ platforms. As part of DFSMS, DFSMSrmm is completely integrated into the IBM storage management strategy. This allows an easier installation and maintenance, as well as standard interfaces with other systems components, such as DFSMSdfp™ and DFSMShsm™.

DFSMSrmm provides a simple and flexible tape management environment, with support for all tape technologies, including IBM automated tape libraries, manual tapes, and other tape libraries.

This IBM Redbook is written for people who are planning to convert from TLMS to DFSMSrmm. We have designed this book to help you with all aspects of the conversion, from the early planning stage through implementation and customization of DFSMSrmm into your production system.

We provide details on the differences between DFSMSrmm and TLMS and compare the terminology, data, and functions. We also explain how to use the IBM-supplied sample conversion programs, validate the converted data, and prepare it for use in a production environment. Working samples that are ready for use are included.

Storage administrators can use this redbook to help plan and execute the conversion. It provides sufficient information to prepare a plan for conversion and then goes through that plan step-by-step.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Mary Lovelace is a Consulting IT Specialist at the International Technical Support Organization. She has more than 20 years of experience with IBM in large systems, storage, and Storage Networking product education, system engineering and consulting, and systems support.

Norbert Schlumberger is an IT Architect with IBM Germany. He has 26 years of experience in storage software and storage management for IBM and customer systems, including 14 years of experience with DFSMSrmm. He has developed, delivered, and taught DFSMSrmm education to customers around the world. He

© Copyright IBM Corp. 2004. All rights reserved.

has many tools available, for example, a Tape Copy Tool to support the DFSMSrmm business. Norbert's areas of expertise include performing conversions from vendor tape management products to DFSMSrmm, and new DFSMSrmm implementations. Norbert provides marketing support for DFSMSrmm, including IBM 3494/IBM 3495 ATLs and VTSs. He has worked at IBM for 30 years.

Thanks to the following people for their contributions to this project:

Robert Haimowitz
International Technical Support Organization, Raleigh Center

Mike Wood
IBM United Kingdom

Charlie Burger
Terry Wiley
IBM San Jose

Sue Hamner
Max Kawano
Jerry Diligiro
IBM United States

Andreas Henicke
Reinhard Siegel
IBM Germany

Yolanda Cascajo Jimenez
IBM Spain

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Archived

Comments welcome

Your comments are important to us!

We want our Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. QXXE Building 80-E2
  650 Harry Road
  San Jose, California 95120-6099

Archived

Archived

1

Introduction

In this chapter, we provide a brief introduction to the following topics:

- Data Facility System Managed Storage removable media manager (DFSMSrmm) basic concepts and functions
- Conversion process overview

At the end of this chapter you will have a basic understanding of DFSMSrmm and the conversion process.

This book is valid for z/OS V1.3 DFSMSrmm and z/OS V1.5 DFSMSrmm. For releases prior to z/OS V1.3 DFSMSrmm, use the redbook *Converting to Removable Media Manager: A Practical Guide,* SG24-4998.

© Copyright IBM Corp. 2004. All rights reserved.

1.1 DFSMSrmm overview

DFSMSrmm is a full function tape management system available as a functional component of z/OS.

DFSMSrmm enables you to manage your removable media as an enterprise library across systems that can share disk. DFSMSrmm can manage all of your tape volumes and the data sets on those volumes. It protects tape data sets from being accidentally overwritten, it manages the movement of tape volumes between libraries and vaults over the life of the tape data sets, and it handles expired and scratch tapes, all according to policies that you define. DFSMSrmm also manages other removable media that you define to it; for example, it can record the shelf location for optical disks and track their vital record status. DFSMSrmm does not yet automatically record information for optical volumes.

1.1.1 Environment

All data set, volume, and policy information is kept in the DFSMSrmm control data set (CDS). The CDS is a VSAM key-sequenced data set (KSDS) that contains all inventory information, and can be either an extended format (EF), or a non-extended format VSAM data set. The only other dedicated data set is the DFSMSrmm journal. This is a sequential data set that is used to record all updates to the CDS. To tailor your processing options, create the EDGRMMxx PARMLIB member. Figure 1-1 shows the DFSMSrmm environment.

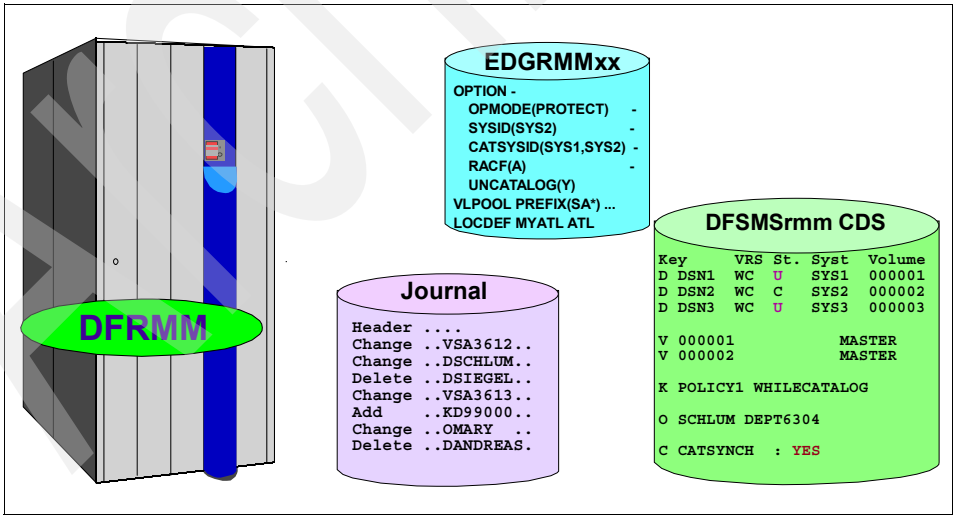


Figure 1-1 DFSMSrmm environment

Note: The benefits available for extended format (EF) VSAM data sets include:

- Data striping
- Data compression
- VSAM extended addressability
- Partial space release
- System-managed buffering
- CDS above 4 GB

All DFSMSrmm interactions take place under control of the subsystem. Utilities are provided to start the housekeeping function, so that you can schedule DFSMSrmm work through your production scheduling systems. One benefit of running as a subsystem is evident in the support for system-managed tape: DFSMSrmm uses a subsystem-to-subsystem interface to provide as much support as possible without involving the operator.

DFSMSrmm supplies ISPF panels as well as TSO commands. Thus, almost anything you can do online, you can also do in batch through TSO command execution.

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

An application program interface (API) is provided to enable user-written programs to access the DFSMSrmm subsystem, and allows interrogation and update of the subsystem.

Use the EDGHSKP utility, with the DFSMSrmm subsystem active to run inventory management activities, which include:

- **Vital record processing**, to determine which data sets to retain and which volume moves are required, based on VRSs
- **Expiration processing**, to identify volumes ready to be released and returned to scratch
- **Storage location management processing**, to set a destination for a volume. Optionally run storage location management to assign shelf locations in storage locations for volumes that are being sent out or returned to the removable media library
- **Backing up the CDS and the journal**, and clearing the journal
- **Creating an extract data set** for report generation

EDGHSKP also has some trial run and predictive features that allow a trial-run of inventory management vital record processing without making any changes to the control data set or journal.

Archive

Use the EDGUTIL utility to create, update, and verify the CDS. Use the EDGBKUP utility to back up and recover the CDS and journal. Both EDGUTIL and EDGBKUP can be executed independently of the subsystem.

CDS backup and recovery

When backup is performed using the DFSMSdss™ DUMP command, DFSMSrmm backs up the CDS and then the journal. DFSMSrmm resets the journal data set and discards journal records if the back up completes successfully. To restore the CDS using the latest CDS backup, both the latest journal backup and the current journal must be used for forward recovery.

DFSMSrmm clears the journal when the CDS and journal are backed up. Clearing the journal data set prevents the journal from becoming full, and reduces the risk of losing updates to the CDS. Backing up the CDS and the journal using DFSMSdss allows you to write the data directly to tape. Using DFSMSdss concurrent copy allows simultaneous updates to the DFSMSrmm CDS so that other tape activity need not be interrupted.

Reports

You can obtain information and create reports, using:

- DFSMSrmm ISPF dialog, TSO RMM commands, or the API
- EDGAUD, EDGJRPT JCL, and EDGRPTD report utilities
- EDGRRPTE sample REXX exec to create 15 standard reports
- DFSORT™'s ICETOOL utility
- The DFSMSrmm report generator

Searches and lists

You can do an online search using either the DFSMSrmm ISPF dialog or TSO RMM subcommands to create lists of resources and display information recorded in the DFSMSrmm CDS. Here are some examples:

- Operators can create lists of scratch volumes to be pulled for use.
- Tape librarians and system programmers can create lists of software products and the volumes on which they reside.
- General users can create lists of volumes they own, shown in the example in Figure 1-2.

Archive

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- -------
DV0001 MHLRES1         2003/182   2003/182   SHELF    0     MR 0
DV0002 MHLRES1  DV0002 2003/182   2003/182   SHELF    1     MV
DV0003 MHLRES1  DV0003 2003/182   2003/182   SHELF    1     MV
DV0004 MHLRES1  DV0004 2003/182   2003/182   SHELF    1     MV
DV0005 MHLRES1  DV0005 2003/182   2003/182   SHELF    1     MV
DV0006 MHLRES1  DV0006 2003/182   2003/182   SHELF    1     MV
DV0007 MHLRES1  DV0007 2003/182   2003/182   SHELF    1     MV
DV0008 MHLRES1  DV0008 2003/182   2003/182   SHELF    1     MV
DV0009 MHLRES1  DV0009 2003/182   2003/182   SHELF    1     MV
DV0010 MHLRES1  DV0010 2003/182   2003/182   SHELF    1     MV
SEARCH COMPLETE - MORE ENTRIES MAY EXIST
10         ENTRIES LISTED
```

Figure 1-2 List of volumes owned by a single user

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

Report utilities

You can create several types of reports using DFSMSrmm reporting utilities. Use EDGRPTD, EDGJRPT, or the DFSMSrmm report generator to create movement and inventory reports, and EDGAUD to create security and audit reports. EDGRPTD uses the DFSMSrmm extract data set as input. EDGJRPT and the report generator use the extended report extract file. EDGAUD uses system management facility (SMF) records as input.

You can use the reports to:

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

Archived

You can use DFSORT or a similar program to generate a formatted report using the information in the EXTRACT data set created by the EDGHSKP utility. Also, the ACTIVITY file and the EDGJACTP sample JCL can be used to create a report from it. For example, you can produce an extract data set listing all volumes to be used on VM with information about volume owners. Then use DFSORT's ICETOOL utility to sort the information by volume and produce a report, complete with title and header information.

Security

You can choose the authorization levels of users for all DFSMSrmm functions. DFSMSrmm uses the MVS™ system authorization facility (SAF) for its authorization checking. You define DFSMSrmm resources to z/OS Security Server Resource Access Control Facility (RACF®) for use during authorization checking. DFSMSrmm can create volume profiles, change them, and delete them on registration, expiration, or release of volumes. DFSMSrmm optionally supports security profiles in the RACF DATASET and TAPEVOL classes as a way to authorize the use of certain RMM TSO subcommands against data set and volume information defined in the CDS.

You can use the access list that DFSMSrmm provides to set the access list in RACF, as well as for authorization checking on non-RACF systems. To display the DFSMSrmm access list, use the TSO RMM LISTVOLUME subcommand or the DFSMSrmm ISPF dialog volume list function. You can also find the access list in the volume records in the report extract data set.

DFSMSrmm provides automatic security classification through installation specified criteria based on data set names. It provides the following control of classified volumes:

- Audit trail of access and change of status through SMF. This audit trail produces information about RACF user ID, group, and job name.
- Operator confirmation of use
- Erasure of data when a volume is released before it is returned to scratch status.

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

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

Archive

1.1.2 How is DFSMSrmm structured?

In Figure 1-3, you can see the basic DFSMSrmm structure. DFSMSrmm runs in its own subsystem address space inside the z/OS operating system. It communicates directly with the Subsystem Interface (SSI), allowing DFSMSrmm to fully control the tape usage in the system.

A user address space is any task requesting information from DFSMSrmm or any user of tape. For example, this could be a batch job creating a tape data set, or an end user listing owned volumes.

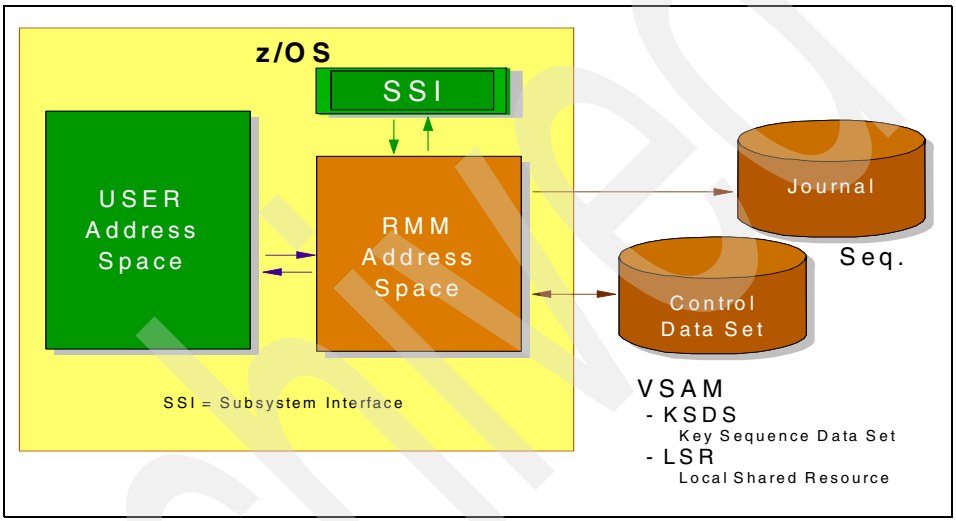


Figure 1-3 DFSMSrmm structure

DFSMSrmm uses the following data sets for its activity:

- Control data set (CDS)
  The control data set is a VSAM key-sequenced data set that contains the complete inventory of the removable media library and storage locations. DFSMSrmm records all changes made to the inventory, such as adding or deleting volumes in the control data set. The control data set should be RACF protected to prevent unauthorized access to the data. DFSMSrmm requires CONTROL access to the control data set.
- Journal
  The journal is a sequential data set, which contains a record of all changes made to the control data set since the last backup.

For information about these data sets, such as structures, size, and definition, refer to 3.1, “DFSMSrmm implementation” on page 60.

Archived

When multiple systems exist, then they can share the same control data set. The control data set cannot be a SYS1 data set if the control data set is to be shared. Figure 1-4 shows the structure in this environment.

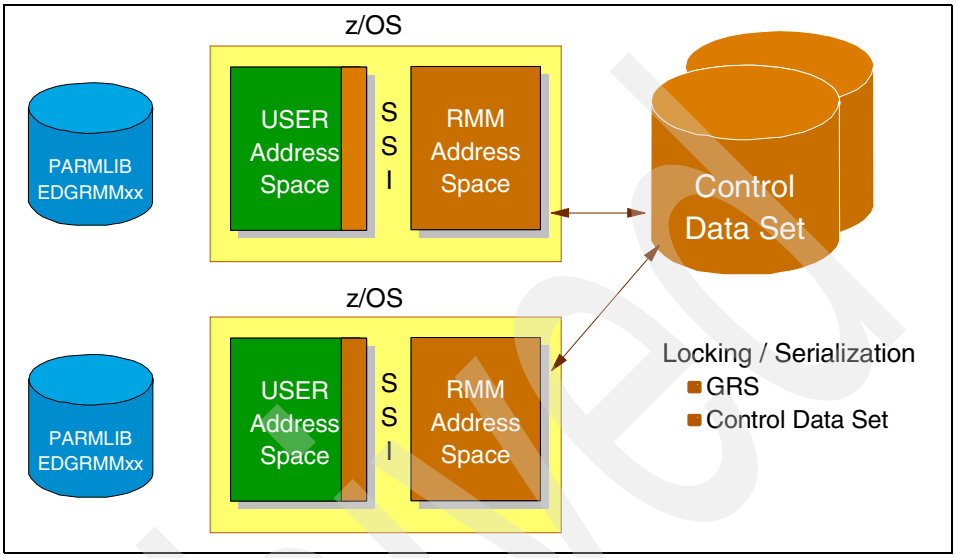


Figure 1-4 DFSMSrmm in a multisystem environment

When multiple systems share one CDS, you can use Global Resource Serialization (GRS) resource definition to solve the possible contention problems. Refer to “GRSRNLxx (optional) - Global resource serialization resourcename list” on page 80 for details on how to customize GRS reserve and release actions for DFSMSrmm.

1.1.3 What are the DFSMSrmm interfaces?

DFSMSrmm interacts with different system and non-system components to perform the tape management functions. We introduce three different types of interfaces: software interfaces, user interfaces, and programming interfaces.

Software interfaces are those interfaces between DFSMSrmm and some system components or activities, such as allocation, open, close, and end of volume events (OCE); and between DFSMSrmm and other software components, such as DFSMSdfp, DFSMShsm, DFSMSdss, RACF, catalog, and OAM.

User interfaces are those that allow the communication between DFSMSrmm and the end user. These user interfaces are the Interactive System Productivity Facility (ISPF) dialogs and TSO commands.

Finally, the programming interfaces are those that allow the communication between DFSMSrmm and user applications, or programs, for being able to get information from DFSMSrmm, or to provide information to DFSMSrmm.

Software interfaces

In Figure 1-5, you can see a general view of what we consider the main DFSMSrmm software interfaces. Using these interfaces, DFSMSrmm either provides support to other users or request services from them. For detailed information about this topic, refer to *DFSMSrmm Implementation and Customization Guide,* SC26-7405.

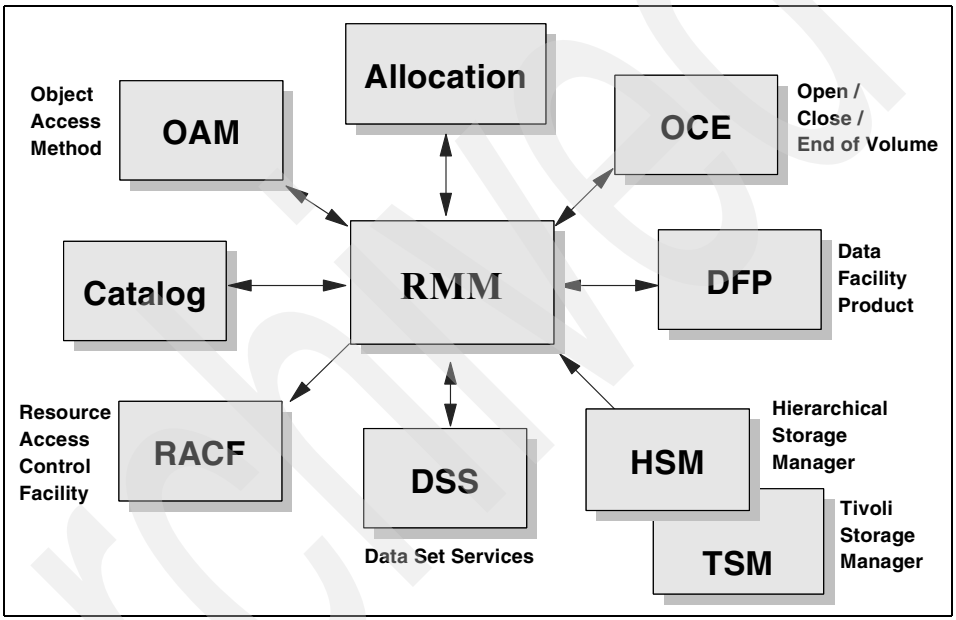


Figure 1-5 Interfaces between DFSMSrmm and other software components

Now we describe what kind of information is exchanged between DFSMSrmm, and each of the software interfaces and the activities performed:

- **Allocation**
  At allocation time, DFSMSrmm intercepts the mount message, assigns a scratch pool, and finds the rack where the volume resides; after that it modifies the mount message to finish the response to this request.
- **Open/close/end of volume interface (OCE)**
  DFSMSdfp provides tape exits for vendor or customer use; DFSMSrmm has the capabilities to work with these exits to manage the open, close, and end of volume event of tape. When such events happen, DFSMSrmm will:

- Validate the mounted tapes.
- Store information in the CDS for volume and data sets if authorized.
- Return block-ID for high speed locate.
- Reject the volume if:
  - A wrong volume is mounted for a specific volume request.
  - A private volume is mounted in response to scratch request.
  - An attempt is made to read a scratch volume.
  - An attempt is made to overwrite a data set and the data set name does not match.
  - An attempt is made to use a specific scratch volume.
  - The first file of the volume does not match what DFSMSrmm has recorded.
  - An attempt is made to read or write a volume using non-standard labels.
  - Unauthorized functions are used, such as bypass label processing, write label, and so on.

- **DFSMSdfp interface**
  DFSMSrmm provides EDGMSGEX exit as an interface to DFSMSdfp. DFSMSrmm uses the IGXMSGEX exit, the MSGDISP installation exit of DFSMSdfp, to call EDGMSGEX to retrieve DFSMSrmm information to update tape drive displays and control the use of cartridge loaders.

  For system-managed tapes, when the volume is open or closed or an end-of-volume event occurs, the SMS ACS routines determine the volume information.

  For non-system-managed tapes, DFSMSrmm calls SMS ACS routines to get information about the storage group for volumes and the management class for data sets.

  Figure 1-6 shows you the differences between these allocations.

Archived

.

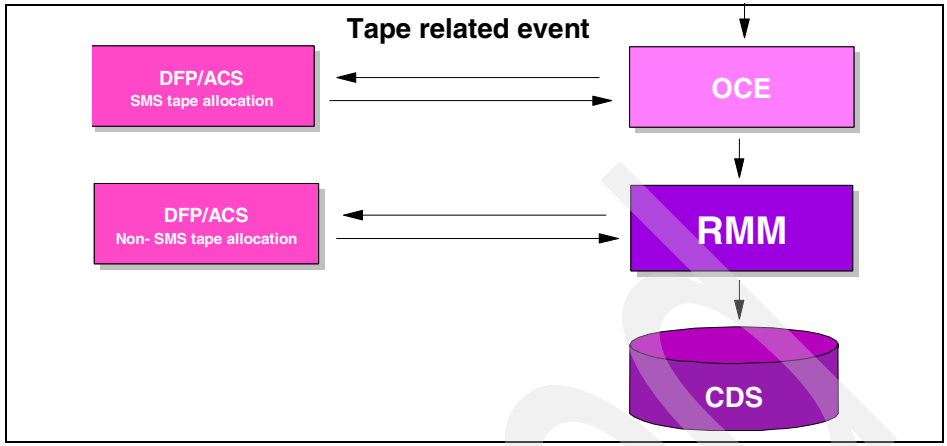


Figure 1-6 SMS tape allocation

- **DFSMShsm and Tivoli® Storage Manager interface**
  DFSMSrmm provides EDGTVEXT and EDGDFHSM programming interfaces for use by products like DFSMShsm and Tivoli Storage Manager when using DFSMSrmm tape management.

  DFSMSrmm treats DFSMShsm like any other user, and retains volumes based on retention period and vital record specifications. DFSMSrmm recognizes when DFSMShsm is opening tape data sets, and for input processing, tolerates the data set names that DFSMShsm uses as long as the last 17 characters of the data set name match.

  For Tivoli Storage Manager (TSM), you should define policies (vital record specifications) to cover the Tivoli Storage Manager tape data sets. A single policy specifying the Tivoli Storage Manager high level qualifier is sufficient to include all Tivoli Storage Manager tape data sets.

- **DFSMSdss interface**
  DFSMSrmm makes use of DFSMSdss user interaction module (UIM) interface to enable the usage of DFSMSdss to backup the CDS. When you use DFSMSdss *concurrent copy* for backup, you minimize the time when DFSMSrmm must prevent updates to the CDS.

- **RACF interface**
  DFSMSrmm does not provide any security functions itself but relies on installed security products, such as RACF or equivalent, to process requests. DFSMSrmm uses the MVS SAF (Security Access Facility) interface to perform authorization checks and other security processing.

  You can use RACF to protect DFSMSrmm resources and to authorize DFSMSrmm functions.

Archived

DFSMSrmm also provides support for using RACF standard tape volume security protection. You can use any combination of RACF TAPEVOL and TAPEDSN options to protect tape volumes as well as tape data sets.

- **Catalog interface**
  DFSMSrmm uses catalog search interface (CSI) to synchronize the control data set with the catalog for tape data sets. It can also exploit catalog exits to mirror catalog updates in the CDS and uncatalog tape data sets according to the definition in PARMLIB.

- **BTLS interface**
  If your installation uses BTLS to drive IBM 3494 and IBM 3495 Automated Tape Library Dataserver operations, you will need to notify BTLS when a volume has to return to SCRATCH status. In a TLMS environment this is done with a batch job that gets information from the TLMS database and then updates the BTLS catalog. In DFSMSrmm you can use a simple REXX CLIST that synchronizes the DFSMSrmm database with the BTLS catalog.

- **SMS interface**
  If you have an IBM 3494 or an IBM 3495 Automated Tape Library Dataserver, you must ensure that it works correctly in the DFSMSrmm environment. You have to change the OAM exits, using the DFSMSrmm version, and provide the correct input to the extract program for those volumes that belong to the tape library.
  DFSMSrmm provides programming interface EDGLCSUX to use the Object Access Method (OAM). The OAM installation exits used by DFSMSrmm are CBRUXCUA, CBRUXEJC, CBRUXENT and CBRUXVNL.

  When one of the following events happens:

  – Change volume use attribute
  – Cartridge eject
  – Cartridge entry
  – Volume-not-in-library

  OAM address space uses the corresponding installation exit to send requests to DFSMSrmm. When DFSMSrmm queries information about library or volumes and does some volume related actions such as eject, delete volume and change volume status, DFSMSrmm uses OAM executable macros like CBRXLCS to send requests to the OAM address space.

  – **Short-on-scratch processing interface**
    DFSMSrmm offers an integrated SMS interface that updates the TCDB automatically.
    In addition, without manual intervention, you can free volumes that are in PENDING RELEASE status when a short-on-scratch condition is detected inside the ATL. Use the PARMLIB option, SCRATCHPROC(RMMSCR), where RMMSCR is the name of the procedure DFSMSrmm starts to replenish scratch volumes in the ATL.

- **High speed locate function**

  DFSMSrmm now records in the control data set the starting and ending tape block IDs for each data set when it is created. With this new technique, DFSMSrmm enables the use of tape block IDs for applications that do not support them (see Figure 1-7).

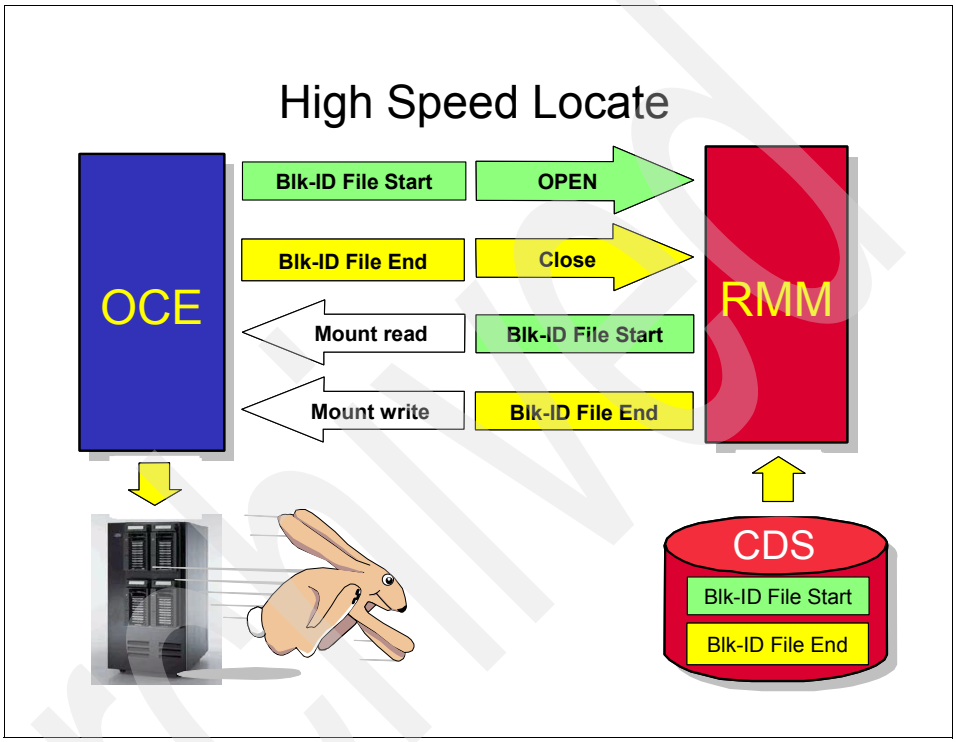


Figure 1-7 High Speed Locate overview

The benefit of using this function is that it enables high speed positioning when there is a large number of data sets in a high capacity tape cartridge.

To gain benefit from this function, all you need is OS/390 Version 2 Release 10 and DFSMSrmm. The system and DFSMSrmm will automatically record and retrieve block IDs, and then use these for high speed data set positioning whenever it is possible.

User interfaces

DFSMSrmm user interfaces allows the user to interact with DFSMSrmm for performing any of the actions related to the tape management. The user can interact with DFSMSrmm in two different ways: ISPF panels, and TSO commands.

Archive

- ISPF dialogs

  The DFSMSrmm ISPF dialog is provided in the ISPF environment, and you can also tailor it to meet your requirements, such as making DFSMSrmm available on any selection panel you want, controlling the access for different type of users. From the ISPF panels, the user can perform most commonly used DFSMSrmm functions, such as define, delete, display volumes, policies, data sets.

- TSO commands

  DFSMSrmm allows you to use a TSO command to define resources, manage resources and create reports. Example 1-1 shows a sample command input for adding a new VRS.

Example 1-1 DFSMSrmm TSO command interface

```
rmm addvrs dsn('ycjres4.test1.**') days count(1)
```

Application Programming Interface

DFSMSrmm provides an Application Programming Interface (API) that allows customers to write programs to obtain services from DFSMSrmm. Figure 1-8 shows the relationship between DFSMSrmm API and other software components.

Using the API, you can issue any of the RMM TSO subcommands from an assembler program.

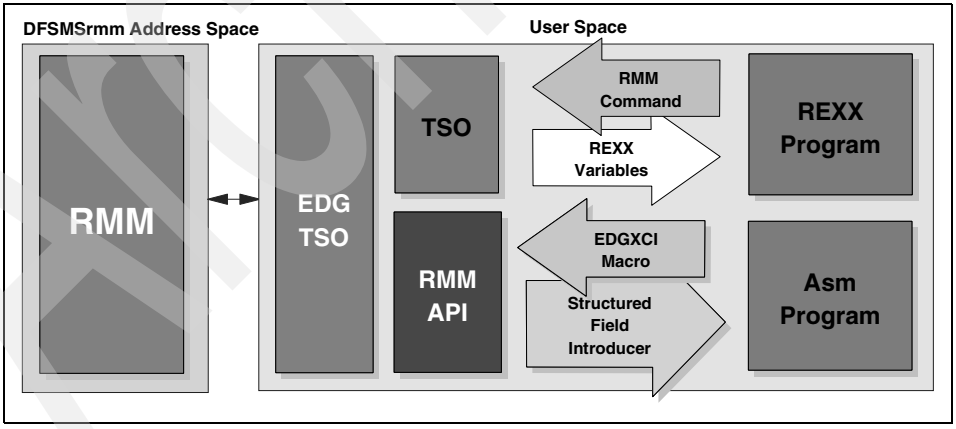


Figure 1-8 DFSMSrmm API

You can use the API to obtain information about DFSMSrmm resources and use the data to create reports or to implement automation. To use the DFSMSrmm API, you must have High Level Assembler installed on your system, and code

assembler language programs. For more information about the API, refer to *DFSMSrmm Application Programming Interface*, SC26-7403.

1.1.4 Library and storage location management

You decide where to store your removable media according to how often the media is accessed and for what purpose it is retained. For example, you might keep volumes that are frequently accessed in an automated tape library (ATL), and you probably use at least one *storage location* or vault to retain volumes for disaster recovery and audit purposes. You might also have locations where volumes are sent for further processing. These locations could be data centers within your company or customer and vendor sites.

DFSMSrmm manages the following:

- **Removable media library**, which contains all tape and optical volumes that are available for immediate use and includes the shelves where they reside. A removable media library usually includes other libraries, such as:
  - System-managed tape libraries; for example, the IBM 3494 and IBM Virtual Tape Server (VTS) Automated Tape Library Dataserver models, and manual tape libraries.
  - Non-system-managed tape libraries, or traditional tape libraries.
- **Storage locations** that are not part of the removable media library because their volumes are not generally available for immediate use. Storage locations are typically used to store removable media that are kept for disaster recovery or vital records.
- **Loan locations**, which are used to track volumes sent to third parties.

Archived

Figure 1-9 shows the relationship of the libraries and storage locations.

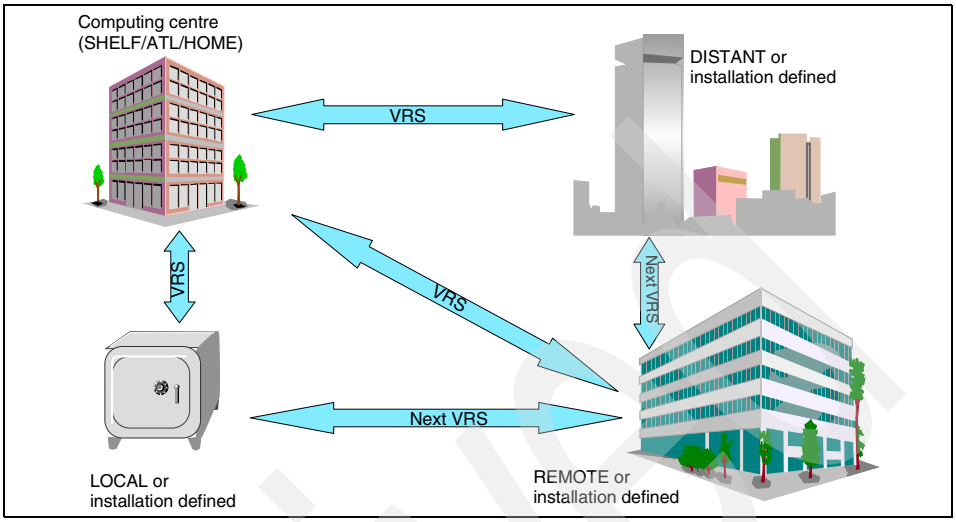


Figure 1-9 Removable media libraries and storage locations

Removable media library

A *removable media library* contains all the tape and optical volumes that are available for immediate use, including the shelves where they reside.

A removable media library usually includes other libraries: *system-managed libraries* such as *automated or manual tape libraries*; and *non-system-managed libraries*, containing the volumes, shelves, and drives not in an automated or a manual tape library.

In the removable media library, you can store your volumes in shelves, where each volume occupies a single *shelf location*. This shelf location is referred to as a *rack number* in the TSO RMM subcommands and in the DFSMSrmm ISPF dialog.

A rack number matches the volume's external label. DFSMSrmm uses the volume serial number to assign a rack number when adding a volume, unless you specify otherwise. The format of the volume serial you define to DFSMSrmm must be one to six alphanumeric characters. The rack number must be six alphanumeric or national characters. This information is illustrated in Figure 1-10.

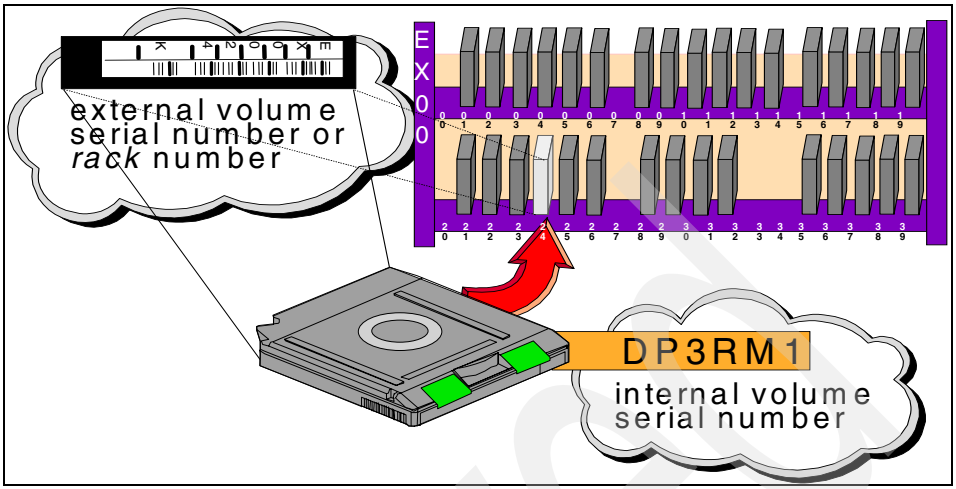


Figure 1-10 Internal labels, external labels, and rack numbers

System-managed tape library

A system-managed tape library is a collection of tape volumes and tape devices, defined in the tape configuration database (TCDB). The TCDB is an ICF catalog of type VOLCAT. It is a new SMS control data set containing tape library and tape volume records.

A system-managed tape library can be either automated or manual. You can have several automated or manual tape libraries. Use an installation-defined library name to define each ATL or MTL to the system, and DFSMSrmm treats each system-managed tape library as a separate location or destination:

- **Automated tape library**

  An *automated tape library* consists of robotics components, cartridge storage areas (or shelves), tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. DFSMSrmm provides support for the automated IBM 3494 Tape Library Dataserver, and also the IBM 3495 Tape Library Dataserver.

  The IBM 3494 Tape Library Dataserver supports 3490E and 3590 tape subsystems. The IBM 3495 Tape Library Dataserver supports 3490, 3490E, and 3590 tape subsystems. The IBM 3494 Tape Library Dataserver can also include the 3494 Virtual Tape Server (VTS). DFSMSrmm supports the IBM Peer-to-Peer VTS by ensuring that you cannot use the names of distributed VTS libraries with DFSMSrmm. You must only use the names of consolidated libraries with DFSMSrmm.

Archive

In Figure 1-11 you can see an overview of the system managed tape environment.

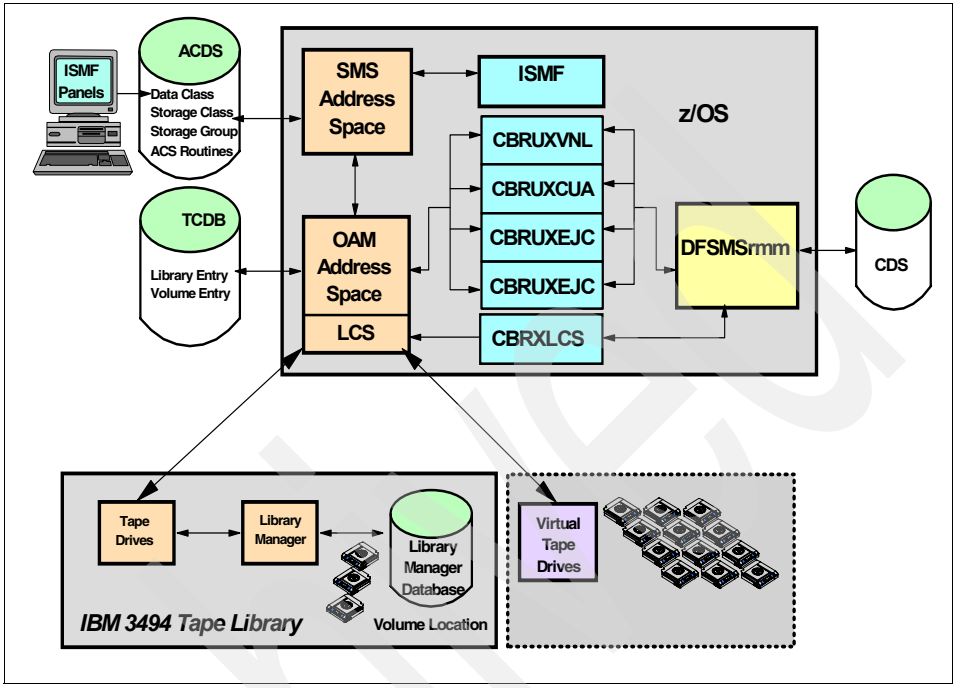


Figure 1-11 System managed tape overview

- **Manual tape library**

  A manual tape library (MTL) is an installation-defined set of tape drives and a customer-defined set of tape volumes with mount capability on those tape drives.The volumes can be physically stored in shelf storage located near the MTL, but since these volumes are specifically defined as residing in the MTL, they are known as library-resident volumes. When the volumes are logically ejected from the MTL, they become shelf-resident volumes. In an MTL environment, the operator or tape librarian responds to commands at the MVS console, manually loading and unloading the tape cartridges.

  Before a tape cartridge can be used, the tape cartridge must first be logically entered into an MTL. Cartridges can be entered into an MTL through invocation of the CBRXLCS manual cartridge entry (MCE) general use programming interface, or through invocation of the LIBRARY ENTER command.

  DFSMSrmm can initiate the MCE process. DFSMSrmm uses MCE processing to add information about a volume residing in an MTL into the TCDB. The **`RMM ADDVOLUME`** or **`CHANGEVOLUME`** subcommands with the

MEDIATYPE operand can be used to define the volume to the MTL. You must ensure that DFSMSrmm has the correct media type for the volume to avoid allocation and mount problems when the volumes are used (refer to Figure 1-13 on page 22).

Non-system-managed tape library

A non-system-managed tape library encompasses all the volumes, shelves, and drives not in an automated or manual tape library. You might know this library as the *traditional* tape library. DFSMSrmm provides complete tape management functions for the volumes and shelves in this traditional tape library. DFSMSrmm defines volumes in a non-system-managed library as being *shelf* resident.

All tape media and drives that the operating system supports are supported in a non-system-managed tape library environment. Using DFSMSrmm, you can fully manage all types of tapes in a non-system-managed tape library, including 3420 reels and 3480, 3490, and 3590 cartridge system tapes.

Storage location

A storage location comprises shelf locations that you define to DFSMSrmm. A shelf location in a storage location is identified by a *bin* number. DFSMSrmm manages two types of storage locations: installation-defined storage locations and DFSMSrmm built-in storage locations. For more information about built-in storage locations, refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405, Chapter 6 "*Managing Storage Locations*."

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

You can now define that a storage location is a home location, allowing volumes to be scratched while in that location. You can use a storage location defined as a home location for all volumes stored in a non-IBM robot tape library.

You can also use the DFSMSrmm built-in storage locations, LOCAL, DISTANT, and REMOTE. Although the names of these locations imply their purpose, they do not mandate their actual location. All volumes can be in the same or separate physical locations.

For example, an installation can have the LOCAL storage location onsite, as a vault in the computer room, the DISTANT storage location can be a vault in an adjacent building, and the REMOTE storage location can be a secure facility across town. An installation defined storage location of SHELF can be onsite,

and used for volume movement within a library to make better use of the library. Because DFSMSrmm provides shelf management for storage locations, storage locations can be managed at the shelf location level. Optionally, you can define storage locations that are not shelf-managed.

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

HOME location

Starting with z/OS V1.3 DFSMSrmm you can define any location as a home location for your volumes. To do this you have to specify each location in your EDGRMMxx PARMLIB member as shown in example Example 1-2.

Example 1-2 Defining a storage location as home location

```
LOCDEF LOCATION(EAST) TYPE(STORAGE,HOME).
```

When you identify a storage location as a home location, you can manage volumes in a storage location like volumes that reside in a LIBRARY location. You can:

- Schedule release actions for volumes when they return to their storage home location
- Return volumes to scratch status while they reside in their storage home location

Figure 1-12 shows you the chances made between z/OS V1.3 DFSMSrmm and previous DFSMSrmm releases. In previous DFSMSrmm releases, the home location was always called SHELF if it was not an automated or manual tape library.

Archived

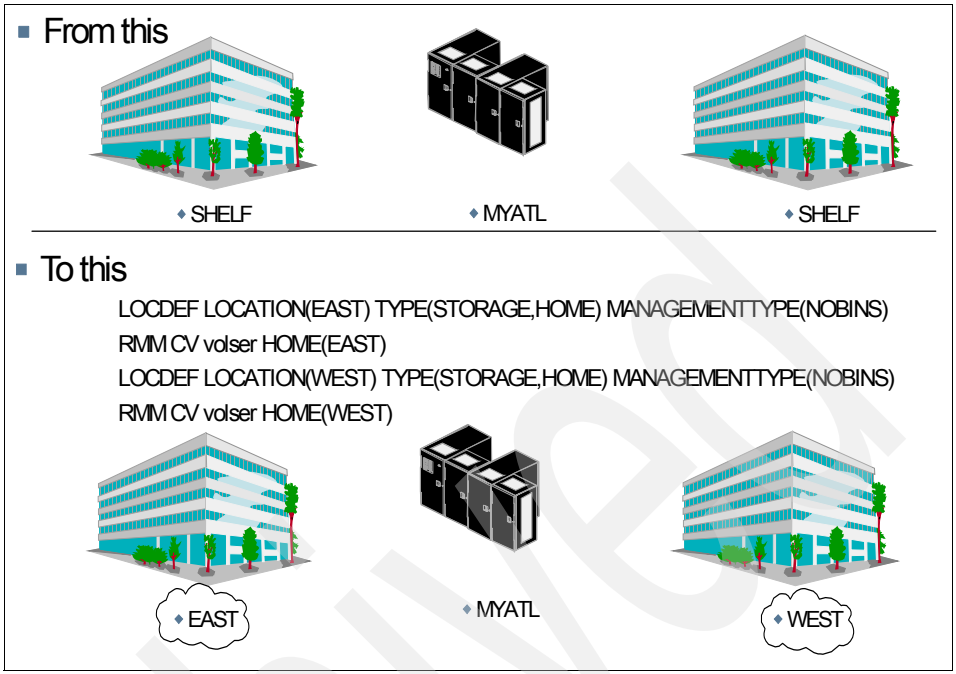


Figure 1-12 Support of different home locations

1.1.5 Media types

The media type specifies the type of physical media of the volumes defined to DFSMSrmm.

DFSMSrmm uses the volume's media type as defined to DFSMSrmm when requesting manual cartridge entry; you must ensure that DFSMSrmm has the correct media type to avoid allocation and mount problems when the volumes are later used. You use the RMM ADDVOLUME or CHANGEVOLUME subcommands with the MEDIATYPE operand to set the media type.

Figure 1-13 shows the different media types and the terms used in SMS and DFSMSrmm.

Archived

Cartridge Type	CST 3480/3490	CST 3490	ECCST 3490	HPCT 3590	HPCT 3590	HPCT 3590	EHPCT 3590	EHPCT 3590	EHPCT 3590	MEDIA5 3592
Tracks	18	36	36	128	25	384	128	256	384	EFMT1
Native Compacted	200 MB 600 MB	400 MB 1.2 GB	800 MB 2.4 GB	10 GB 30 GB	20 GB 60 GB	30 GB 90 GB	20 GB 60 GB	40 GB 120 GB	60 GB 180 GB	300 GB 900 GB
ATL Media Label	1 or blank	1 or blank	E	J	J	J	K	K	K	JA
OAM Media Type	01	01	02	03	03	03	04	04	04	05
SMS Tape Category	MEDIA1	MEDIA 1	MEDIA2	MEDIA3	MEDIA3	MEDIA3	MEDIA4	MEDIA4	MEDIA4	MEDIA5

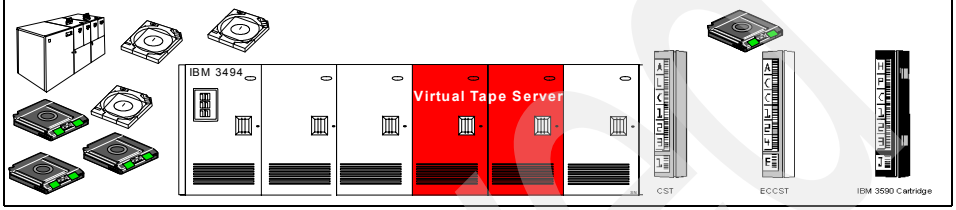


Figure 1-13 Different cartridge media types

Figure 1-14 shows how you can use the different cartridges on all the available tape drives.

Cartridge Type	CST	CST	ECCST	HPCT	HPCT	HPCT	EHPCT	EHPCT	EHPCT	MEDIA5
Tracks	18	36	36	128	256	384	128	256	384	EFMT 1
3480 without IDRC	read & write	rewrite*	-	-	-	-	-	-	-	-
3480/3490 with IDRC	read & write	rewrite*	-	-	-	-	-	-	-	-
3490E writes 36 tracks	read or rewrite*	read & write	read & write	-	-	-	-	-	-	-
3590 B1x writes 128 tracks	-	-	-	read & write	rewrite*	rewrite*	read & write	rewrite*	rewrite*	-
3590 E1x writes 256 tracks	-	-	-	read or rewrite*	read & write	rewrite*	read or rewrite*	read & write	rewrite*	-
3590 H1x writes 384 tracks	-	-	-	read or rewrite*	read or rewrite*	read & write	read or rewrite*	read or rewrite*	read & write	-
3592 J1x writes EFMT1	-	-	-	-	-	-	-	-	-	read & write

Figure 1-14 Tape drive matrix

Note: To reformat 384-track format cartridges to 128-/256-track format, you need following EC levels:

3590B = D19328

3590E = F23079

* When rewriting a tape in a format different from the current format

1.1.6 Volume management

DFSMSrmm manages volumes based on the volume type, volume status, and the volume media.

Types of volumes

DFSMSrmm provides support for physical, logical, and stacked volumes (see Figure 1-15).

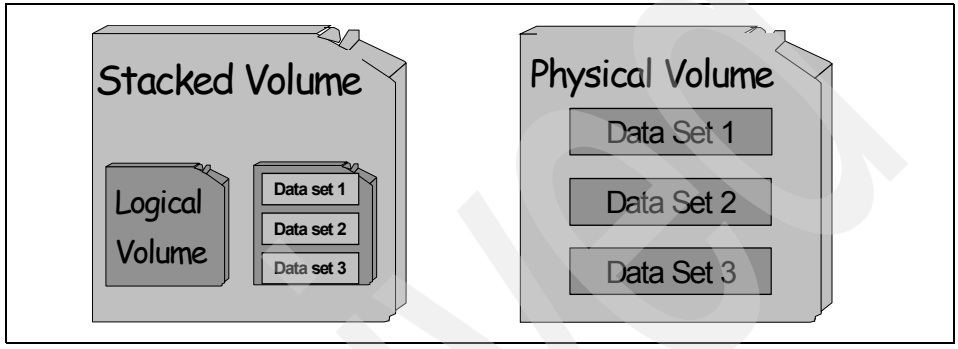


Figure 1-15 Stacked, logical, and physical volumes

- Physical volumes

  DFSMSrmm provides support for all types of physical volumes, tracking their use and the data they contain, and providing vaulting and retention services according to policies assigned to the data sets on the volumes.

- Logical volumes

  These reside in a Virtual Tape Server (VTS) or on exported stacked volumes. DFSMSrmm tracks the use of logical volumes and the data that resides on the volumes. Retention services are provided according to policies assigned to the data sets on the volumes. DFSMSrmm supports exporting of logical volumes on stacked volumes and provides vaulting services for these stacked volumes based on the policies assigned to the data sets on the contained logical volumes. Logical volumes can be removed from a VTS using export processing.

- Stacked volumes

  These have a one-to-one association with physical tape media and are used in a VTS to store logical volumes. Stacked volumes are not used by MVS applications, they are used by the VTS and its associated utilities. Stacked volumes can be removed from a VTS to allow transportation of logical volumes to a vault or to another VTS.

Archived

DFSMSrmm has added support for stacked volumes to allow identification of the stacked volumes in a Virtual Tape Server (VTS) and direct management of these volumes when exported. These stacked volumes are assigned to specific slots when moved to storage locations that require shelf management.

DFSMSrmm provides support for importing logical volumes on stacked volumes either into the VTS from which it was exported, or into a different VTS.

If stacked volume support is enabled, you can define stacked volumes and the logical volumes associated to them using the RMM ADDVOLUME subcommand in Example 1-3.

Example 1-3 RMM ADDVOLUME subcommand example

```
RMM AV convol TYPE(STACKED) LOCATION(SHELF)
RMM AV volser TYPE(LOGICAL/PHYSICAL) STATUS(MASTER) CONTAINER(convol)
```

You also can use stacked volume support as a way to manage the boxes in which you transport and store tapes. You can associate volumes to a box, by defining the box as a stacked volume, and using RMM TSO subcommands to add the volumes to the box as the CONTAINER.

Volume status

Within DFSMSrmm a volume can be in one of six different statuses:

Scratch Where the volume is available for use. That is a volume that does not contain valid data.

Master The status assigned to a volume at open time after a non specific mount.

User Status assigned to a volume after a user get volume request or assigned this status through an DFSMSrmm command.

Pending release This is a transient status that occurs after a master or user volume expires and prior to is becoming scratch.

Init The volume is a scratch volume waiting to be initialized. Once initialized, the status changes to SCRATCH.

Entry The volume is a scratch volume coming into an automatic tape library (ATL). Once entered, the status changes to SCRATCH or INIT.

Figure 1-16 shows the life cycle of a volume.

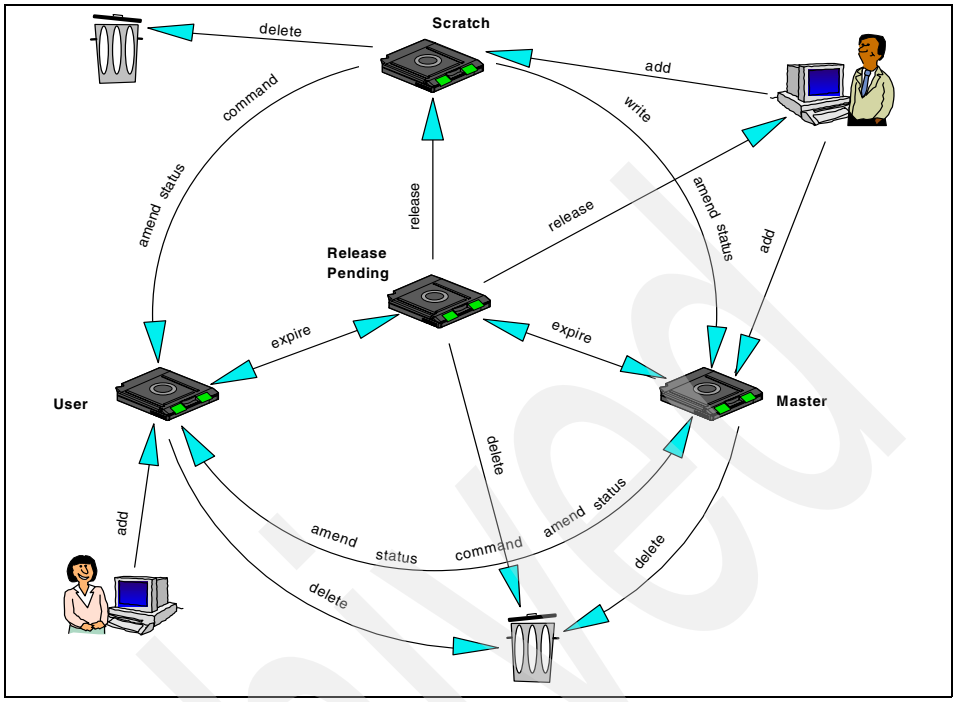


Figure 1-16 Life cycle of a volume

Most of the volumes that DFSMSrmm manages in your installation are probably *scratch* volumes, that is, volumes that are used again and again by different users. Each time a volume is used, it is retained and managed by policies that you define to DFSMSrmm. When the data is no longer required, the volume is returned to scratch status, and is ready for use by another user.

With DFSMSrmm, you can add or delete volumes without stopping your normal tape activities.

In addition to managing scratch volumes, DFSMSrmm can manage any volume you define to it, including those that you treat as *foreign volumes* under your existing system. DFSMSrmm supports any volume serial (VOLSER). You can still process volumes as foreign to DFSMSrmm by either not defining them to DFSMSrmm or using the DFSMSrmm *ignore* support through the EDGUX100 exit.

DFSMSrmm records information for data sets on all files of a tape volume. It also can manage the RACF TAPEVOL and/or TAPEDSN profiles for tape volumes and uncataloged data sets. Figure 1-17 shows how DFSMSrmm processes records related to tape data sets.

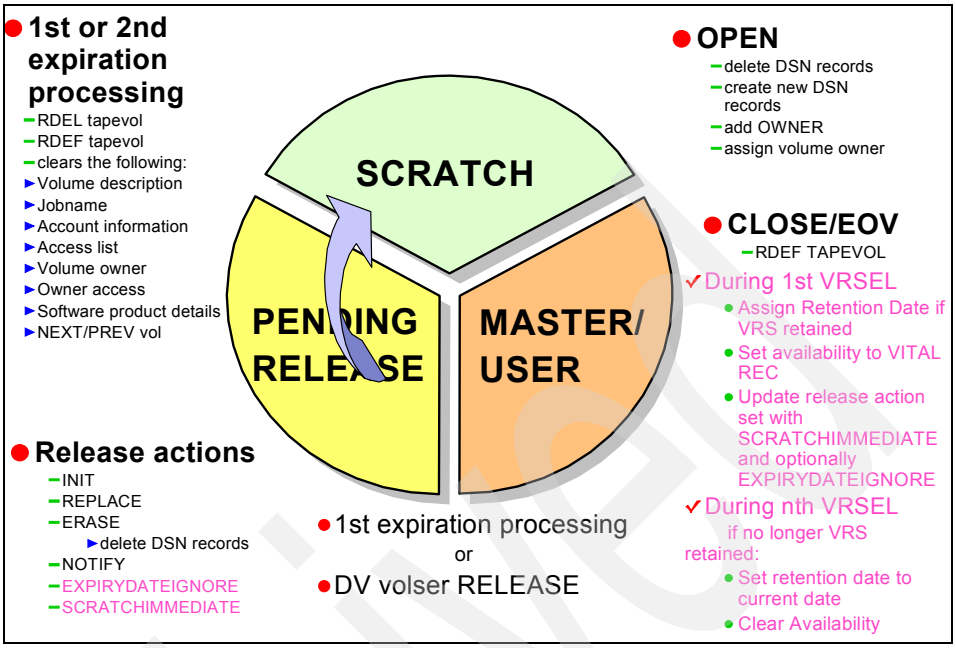


Figure 1-17 Timing of related events for a volume

Validation

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

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

- Changes the volume status from scratch to master for nonspecific mount requests
- Sets an expiration date for the volume and ensures that the maximum expiration date is not exceeded (repeated at close time) as defined in the PARMLIB member.
- Records information about data sets on the volume (the data set name is recorded at open time; all other information is recorded at close time)
- Counts the number of times a volume is used
- Counts the number of temporary and permanent errors encountered
- Sets a security classification (as defined in the DFSMSrmm PARMLIB) based on the data sets that reside on the volume

Archive

- Prevents reading of data on a volume in scratch status when DFSMSrmm is running in protect mode.

All of the above actions occur at open time, except where noted.

Using EDGUX100 to ignore volume serial numbers

DFSMSrmm can ignore volumes regardless of whether the volumes were defined to DFSMSrmm or not.

To request that DFSMSrmm ignore a volume, tailor the EDGUX100 DFSMSrmm installation exit to use undefined volumes or duplicate volumes. When you use the EDGUX100 installation exit, DFSMSrmm calls the exit each time a volume is opened. The sample installation exit checks the JCL–specified EXPDT value for the special date 98000 or for the ACCODE value *x*CANORES. If the 98000 special date or the ACCODE value *x*CANORES is found, the EDGUX100 exit uses the installation exit parameter list to request that DFSMSrmm ignore the volume. Figure 1-20 shows you how the EDGUX100 user exit works. This support was introduced with APAR OW53763.

To request that DFSMSrmm ignore a volume, you also have to define a RACF FACILITY class entity as shown in Figure 1-18.

```
STGADMIN.EDG.IGNORE.TAPE.volser
```

Figure 1-18 Global ignore volume RACF resource

If you want to distinguish between volumes managed by DFSMSrmm and volumes not managed by DFSMSrmm, define the RACF profiles as shown in Figure 1-19 to check if the user is authorized to request that DFSMSrmm ignore the volume.

```
STGADMIN.EDG.IGNORE.TAPE.RMM.volser
STGADMIN.EDG.IGNORE.TAPE.NORMM.volser
```

Figure 1-19 RACF ignore volume resources

In the RACF profile *volser* is the volume serial number of the mounted volume or requested tape volume. Figure 1-20 shows how the BYPASS DFSMSrmm function works.

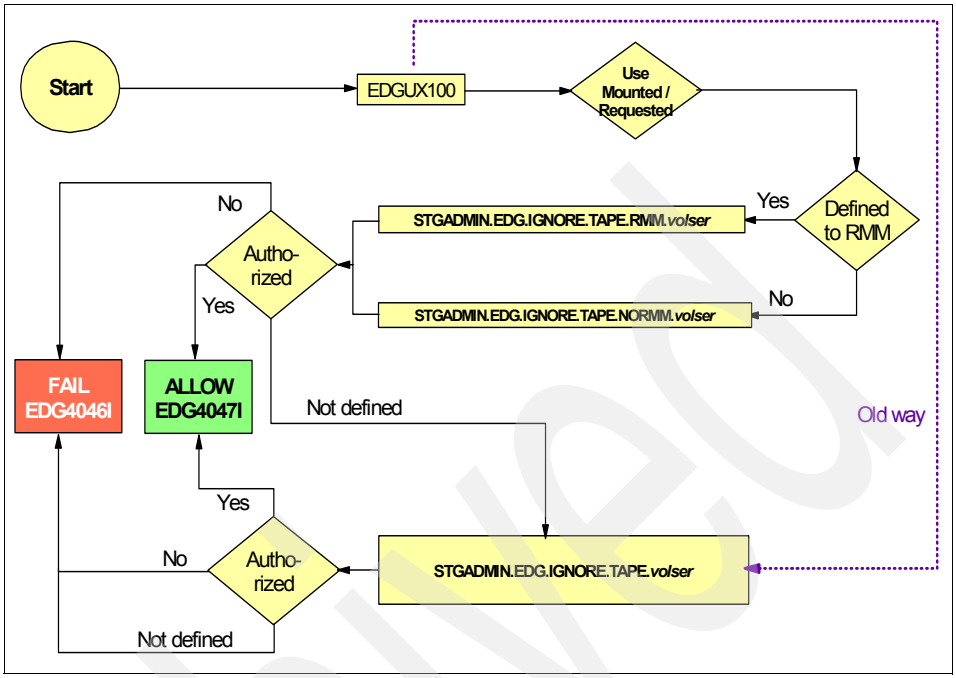


Figure 1-20 Bypass DFSMSrmm function

If you have defined the *RMM* and *NORMM* ignore profiles, DFSMSrmm will make the authorization checks as shown in Table 1-1.

Table 1-1 DFSMSrmm ignore profile checking

Authorization check	Profile used
Volser not found in CDS	NORMM
Volser found in CDS and match on HDR1 17 character data set name	RMM
Volser found in CDS and no match on HDR1 17 character data set name	NORMM

Support for duplicate volumes

In DFSMSrmm, volume serial numbers are used to identify volumes and to identify the volume label. DFSMSrmm allows you to define a volume using a serial number that is different from the one that is recorded in the volume label. This way you can define volumes with duplicate volume serial numbers.

DFSMSrmm open processing has been updated to detect duplicate volumes and to use the correct CDS volume information to manage the volume. You can

Archived

continue to have duplicate volumes ignored, but the recommendation is to add them to the DFSMSrmm CDS and manage them. Figure 1-21 shows you how DFSMSrmm will support duplicate volume serial numbers by specifying VOLSER, RACK, and the new VOL1 operand of the ADDVOLUME function.

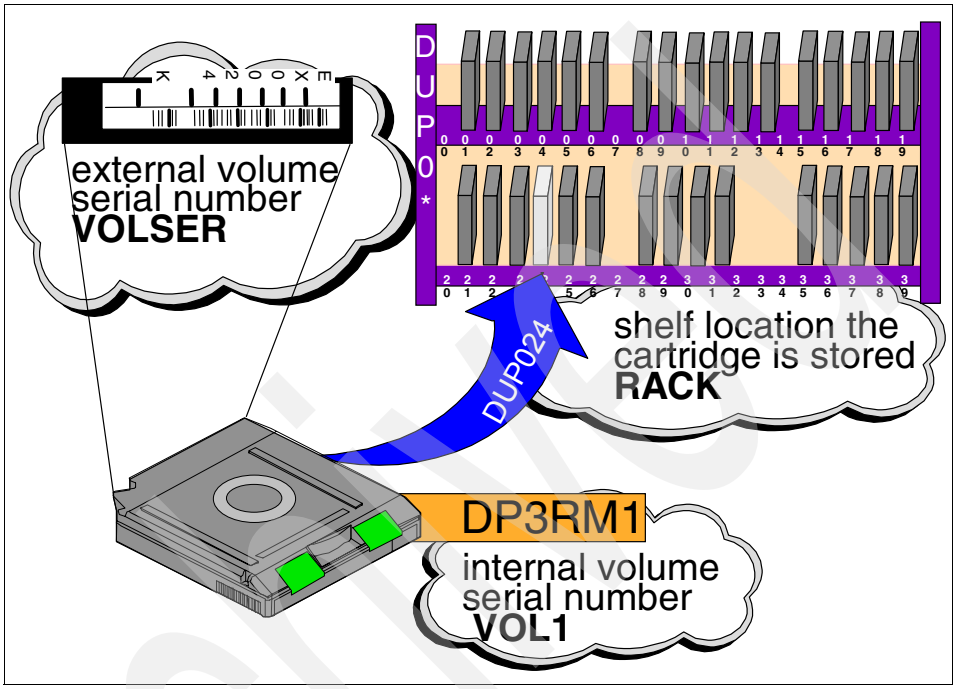


Figure 1-21 DFSMSrmm duplicate volume support

Each volume defined to DFSMSrmm must be defined using a unique volume serial number, but for private physical volumes you can also define the VOL1 label volser. You can have any number of duplicate volumes with the same VOL1 label volser. You can use duplicate volumes for both input and output processing, and do not need to use BLP or DFSMSrmm ignore processing. You do not need to enable duplicate volume support. You just start to define and use duplicate volumes when you need the support. You use the DFSMSrmm ADDVOLUME or CHANGEVOLUME subcommand with VOL1(volser) to define a duplicate volume.

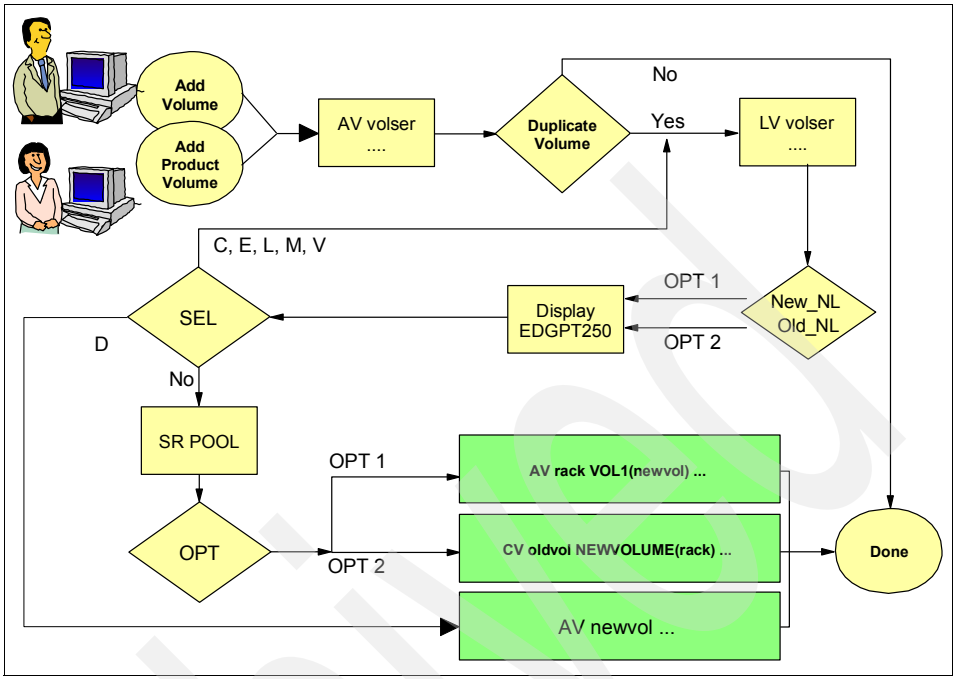


Figure 1-22 Flow chart to add a duplicate volume serial number

A no-label volume cannot be a duplicate volume, because you can define such a volume to DFSMSrmm using any unique value for the volser, and no special processing is required by DFSMSrmm to distinguish the volume from another identical volume. DFSMSrmm is the only component in the system that knows a volume is a duplicate. DFSMSrmm hides this fact from others by substituting the VOL1 label volser read from the volume with the requested volume once DFSMSrmm has validated that the correct volume is mounted. The requested volume is the unique volume serial number you specify in your JCL or allocation request that identifies the volume to DFSMSrmm. DFSMSrmm exploits the capabilities of the Tape Mount exit to check the VOL1 label volser. DFSMSrmm uses the requested volser to find a volume defined to RMM, and validates that the VOL1 label volser matches the VOL1 label volser defined to DFSMSrmm. The user specified volser identifies the volume record to be read from the DFSMSrmm CDS. The VOL1 label volser must match the DFSMSrmm recorded VOL1 label volser, or the volume record has no recorded VOL1 label volser, and the requested volume matches the VOL1 label volser.

Note: In DFSMSrmm we have three different values to identify a volume:

VOLSER	To specify the volume serial number
VOL1	To define a duplicate volume when the VOL1 label volume serial number does not match the volume you are defining to DFSMSrmm
RACK	Specifies a shelf location in the removable media library where DFSMSrmm stores the volume

If you add a new volume with a volume serial number of a volume previously defined in the CDS, DFSMSrmm will display a new panel as shown in Figure 1-23.

```
EDGPT250                  DFSMSrmm Add Duplicate Volume
Command ===>                                              Scroll ===> PAGE

Volume DV0001 duplicates existing volume DV0001
  Enter selected option for volume DV0001 or use line command below
  Any recommended option is pre-selected for you

Option ===>  1
  1 Add the volume as a duplicate
  2 Change the existing volume to be a duplicate

Specify a pool for the duplicate volume
Pool prefix ===>  DUP*

The following line commands are valid: C,D,E,L,M,V
  Volume VOL1   Rack                    Data        Product            Feat
S serial volser number Status   Location Sets  Label number   Level    code
- ------ ------ ------ -------- -------- ----- ----- -------- --------- ----
  DV0001 TST001        MASTER   SHELF    0     SL
```

Figure 1-23 Add duplicate volume

1.1.7 Policies for retention and movement

DFSMSrmm provides policy management for movement and retention at the data set, volume, or volume set level. Every tape data set can have a policy, and each policy can specify movement as well as retention. The retention and movement policies you define to DFSMSrmm are known as *vital record specifications* (VRSs). You use them to indicate how long and where you want to keep data sets or volumes. You also use them to define how volumes are to be moved among the libraries that DFSMSrmm supports and the storage locations defined for vital records and disaster recovery purposes.

Archived

You can create a vital record specification *chain* to cause a sequence of volume moves. The first vital record specification in the chain is a data set name, SMS management class, DFSMSrmm management value or volume vital record specification. These specify one location in which the data set or volume being retained is to be stored. You would add one name vital record specification for each additional location or retention value that applies to the volume or data set.

DFSMSrmm records the starting location for a volume when the volume is initially defined to DFSMSrmm, or when the volume information is changed. This starting location is known to DFSMSrmm as a *home* location. Home is the place from which volumes start, and to which they are returned, when the identified retention and movement actions have been completed.

Data set retention

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

- **Retention by PARMLIB option - VRSEL(NEW|OLD)**
  You can control the extent to which data set name and management value VRSs are used. You can define different retention criteria in a next VRS only if you specify VRSEL(NEW).
- **Retention by cycles**
  You can retain a minimum number of cycles or copies of a data set. This type of retention applies to generation data groups (GDGs) or like-named data sets identified by pseudo-GDG data set names. For non-GDG data sets, DFSMSrmm considers each occurrence of a data set to be a cycle.
- **Retention by BYDAYSCYCLE**
  You can specify to retain all instances of a data set created on a single day as a single cycle.
- **Retention until expired**
  You can retain a data set until volume expiration date has been reached or the retention amount has elapsed.
- **Retention by number of elapsed days**
  You can retain each copy of a data set a given number of days since it was created.
- **Retention by days since last referenced**
  You can retain each copy of a data set for a set number of days since the data set was read or written.
- **Retention while data set is cataloged**
  You can retain any data set as long as it remains cataloged. Catalog status

can also be used in combination with management by cycles, elapsed days, days since last referenced, and volume expiration date.

- **Retention by extra days**
  On a NEXTVRS you can specify 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. This option can only be used on a NEXTVRS and not a primary VRS.
- **Retention to a specific date**
  You can set a deletion date for a VRS. When that data is reached, the VRS is deleted. All data sets and volumes that would match the VRS become eligible for release processing, or they might match a less specific VRS that specifies different retention and movement information.
- **Retention by expiration date**
  You can retain the data set on a volume as long as the volume expiration date has not yet been reached. You can also use a combination of catalog status, volume expiration date, and other retention options to retain data sets.
- **Retention of open data sets**
  You can specify a separate policy to apply to all data sets that are currently open.
- **Retention of data set closed by abend processing**
  You can specify a separate policy to apply to all data sets that were open at the time of an application or system abend.
- **Retention of set**
  This allows you to retain all volumes of a set until all of the data sets on those volumes have expired. Use the EDGRMMxx PARMLIB option RETAINBY(SET).

Figure 1-24 shows you an example of a multi-file/multi-volume set. The first and second data set are not expired, but the third one has since yesterday. The third volume of the set is only retained if PARMLIB option RETAINBY(SET) is specified.

Archived

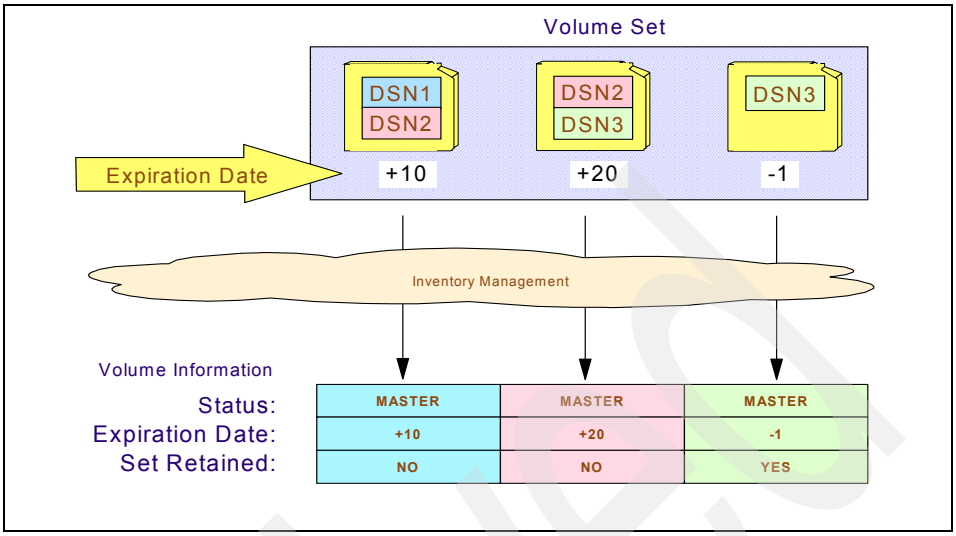


Figure 1-24 RETAINBY(SET): Expiration date

Your installation can also define a VRS management value or a SMS management class to define retention and movement policies. If you have specified VRSEL(NEW) in the EDGRMMxx PARMLIB member, you can have up to two policies per data set: one policy using a data set name mask, and one using VRS management class or VRS management value. The definition of VRSEL(OLD) allows only one policy for each data set.

A VRS management value assigns management and retention values to tape data sets. You can define data set VRSs for VRS management values to provide support for special JCL-specified expiration dates, and to allow DFSMSrmm to manage those data sets with special dates. You can use the sample installation exit EDGUX100, in SAMPLIB, to assign VRS management values.

Volume retention

For a volume, you can request the following types of retention using a specific or generic VOLSER:

- ***Retention by number of elapsed days since creation***
  Which indicates that a volume be retained until the specified number of days since creation have elapsed
- ***Retention by number of volumes***
  Which indicates how many volumes should be retained

SMS ACS support

You can use the ACS routines to process the special calls that DFSMSrmm makes to the SMS subsystem for ACS processing to set a SMS management class. DFSMSrmm requests that the management class and storage group routines are run. The environment variable is set to RMMPOOL, so that you can differentiate allocation requests for system-managed data sets from requests by DFSMSrmm for a storage group.

You can use SMS ACS to decide if data sets are to be system-managed or non-system-managed. Using the SMS pre-ACS calls to DFSMSrmm help you to use your SMS ACS logic. Use the pre-ACS processing for decisions on the MSPOOL and MSPOLICY values that come from DFSMSrmm scratch pool and EDGUX100 exit processing. You can plan to move the EDGUX100 exit decisions into your SMS ACS routines to enable the EDGUX100 exit processing to be ignored or removed some time in the future.

DFSMSrmm attempts to pass values for the MSPOOL and MSPOLICY ACS read-only variables when the PARMLIB option PREACS(YES) is set. If you do not use the EDGUX100 exit, the MSPOOL variable is set to the DFSMSrmm system based scratch pool decision. If you do not use the EDGUX100 exit, there is never a value set for MSPOLICY. DFSMSrmm sets a pre-ACS variable only if the variable has not yet been set using the pre-ACS exit. Since DFSMSrmm gets control after the installation exit, any vendor or customer decisions take precedence (see Figure 1-25).

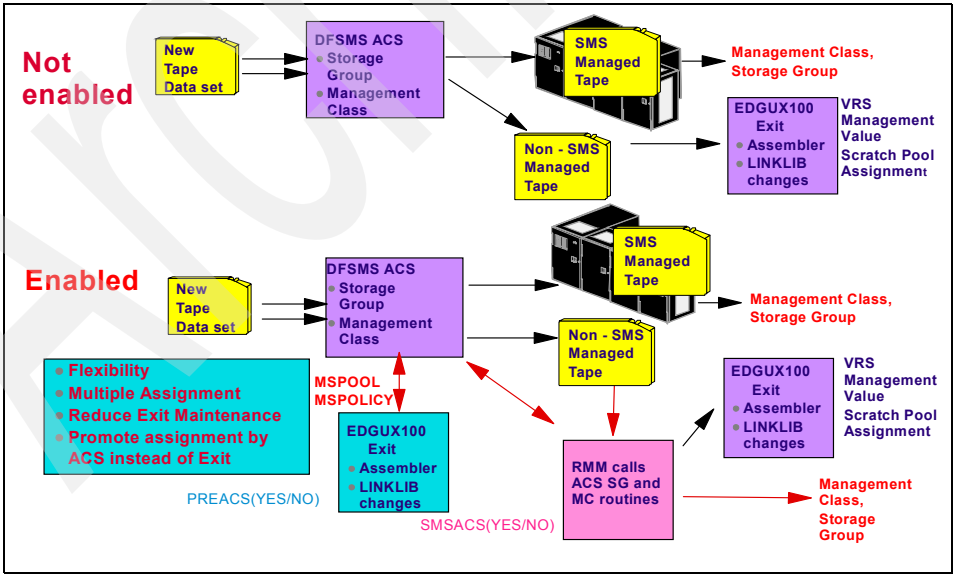


Figure 1-25 DFSMSrmm and SMS ACS support

Volume retention

To define retention and movement policies for volumes, you can use a specific or generic VOLSER. If you use a specific VOLSER, the volume matching that number is retained by days since creation. If you use a generic VOLSER, any volumes matching the generic VOLSER are retained. These volumes must be of user or master status.

Movement

DFSMSrmm records the starting location for a volume when the volume is initially defined to DFSMSrmm, or through a change to the volume information. This starting location is known to DFSMSrmm as a *home* location. Volumes are returned to the home location when the identified retention and movement actions have been completed.

You can give any system-managed library or storage location as a target destination for a volume move. Obviously this location must be defined to DFSMSrmm. The location could be an ATL, MTL, storage location or shelf.

You can also define policies to provide retention information for data sets and volumes that must be moved through multiple locations before they expire. You define such policies by creating a VRS chain.

When multiple data sets on the same volume are retained by VRSs, and each VRS contains a different destination, DFSMSrmm decides where to move the volume according to priority number.

Unlike some other systems, DFSMSrmm does not do storage location management on the basis of the first file on a volume. Storage location management is based on the priority of the storage location management requirements for all data sets on a volume. Thus, DFSMSrmm must sometimes do conflict resolution, and only retains and moves volumes with a data set retained by a VRS.

A recently added function of DFSMSrmm allows you to move a SET of volumes, thus keeping related volumes together. Use the PARMLIB option MOVEBY(SET).

Storage location management with DFSMSrmm is essentially similar to vault management in other tape management systems. What seems to be unique to DFSMSrmm is the concept of volume-in-transit. In other words, volumes have been scheduled for a move or are on their way, but they have not yet been received at the storage location. Operators can tell which volumes are still in transit. A special command moves volumes from in-transit status to storage-location-resident status. Installations that have established special

Archive

storage location names to cover transport time will need fewer storage locations. Installations that do not want librarians issuing this command can automate the confirmation of movement by issuing TSO commands or through REXX procedures.

If you have VRSEL(NEW) specified in the EDGRMMxx member of PARMLIB, you can now have separate movement and retention policies. Use a VRS based on management value or management class name to define retention criteria; use a VRS based on data set name mask for defining movement.

1.1.8 DFSMSrmm automated tape library support

DFSMSrmm excels at automated tape library (ATL) management. In an automated tape library environment, you have different databases to store volume and location information. In the DFSMSrmm CDS, you have the volume serial number, the volume status, and the volume location information. In the SMS tape volume database (TCDB), you have the volume serial number, the volume status, and the location information. In the library manager database, you have the volume serial number and the volume status information. DFSMSrmm will keep all three databases in sync by using all four available OAM installation exits:

CBRUXCUA	Change use attribute exit
CBRUXEJC	Eject cartridge exit
CBRUXENT	Entry cartridge exit
CBRUXVNL	Volume not in library exit

Refer to the manual *DFSMSrmm Implementation and Customization Guide*, SC26-7405, chapter *"Running DFSMSrmm with System-Managed Tape Libraries"* for more details. Figure 1-26 shows you the entire ATL environment.

Archive

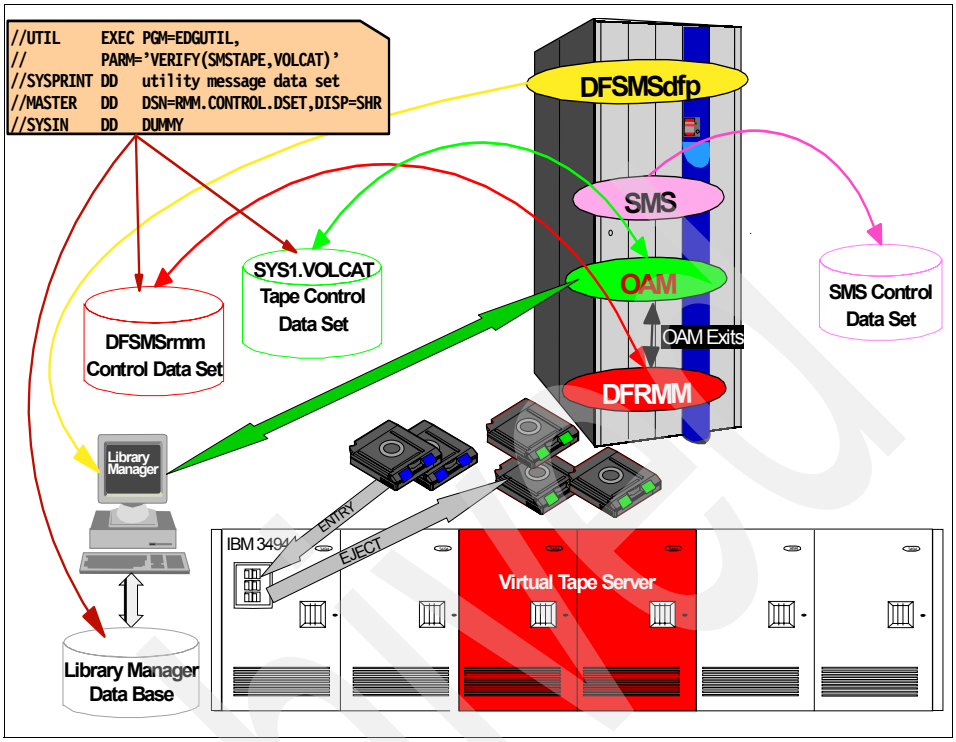


Figure 1-26 Automated tape library (ATL) environment

The components of system-managed tape are:

- OAM and its LCS
- Storage Management Subsystem (SMS)
- Integrated Storage Management Facility (ISMF)
- Installation wide exits
- Tape management system
- Library Manager

DFSMSrmm provides the following support for system-managed tape libraries:

- Cartridge entry processing
- Cartridge eject processing
- Expiration management
- Volume-not-in-library support
- Movement between libraries and storage locations
- Partitioning the libraries with other systems
- VTS import/export processing
- Tracking and managing logical, physical, and stacked volumes

In the following sections we will review some of the items in the list.

Cartridge entry processing

DFSMSrmm automatically adds volumes to the DFSMSrmm control data set during entry processing and when you use volumes. If a volume defined to DFSMSrmm is entered into a system-managed tape library without using the RMM CHANGEVOLUME command to set the volume location to the library name, DFSMSrmm updates the DFSMSrmm control data set volume record with the library name, library type, and volume entry status. DFSMSrmm does not update the home location name for the volume.

DFSMSrmm allows you to partition system-managed tape libraries using the DFSMSrmm REJECT PARMLIB option to identify volumes that you want to partition. Volumes can be identified based on volume prefix and as individual volumes defined to DFSMSrmm on z/OS.

The REJECT parameter of the EDGRMMxx PARMLIB can be used to control cartridge entry processing into the library and whether or not the cartridge is to be used for input processing only. Figure 1-27 shows examples of using the **REJECT** command in the EDGRMMxx member.

```
REJECT OUTPUT(CC12*)
REJECT ANYUSE(VM1*),OUTPUT(VM*)
REJECT ANYUSE(DD0*)
REJECT ANYUSE(*)
```

Figure 1-27 REJECT command examples

Recommendation: Define all volumes to DFSMSrmm before entering them into a library. Define private volumes prior to entry into the automated tape library, and set the ISMF default cartridge entry status to scratch.

Ejecting volumes from system-managed libraries

You can eject physical volumes from a system-managed library by using various methods:

- Using the library manager console commands
- Using ISMF commands
- Using DFSMSrmm commands through ISPF or batch

Archived

Volume-Not-In-Library processing

When a volume is requested for a job, and is not in a system-managed tape library, the DFSMSrmm supplied CBRUXVNL exit is called so the volume can be inserted into a library to prevent job failures from occurring.

Verifying the databases

You can use the DFSMSrmm utility EDGUTIL with the parameters VERIFY(VOLCAT,SMSTAPE) to check the consistency of information in the DFSMSrmm control data set, the tape configuration database (TCDB), and the library manager database. If you have differences, you can use the same utility with a PARM of MEND to correct the errors. Figure 1-26 shows you the entire environment if you have an IBM ATL installed.

VTS import/export processing

DFSMSrmm supports Virtual Tape Server (VTS) libraries such as automated system-managed libraries with extensions to handle the special requirements for different volume types and functional capability. DFSMSrmm supports virtual tape server import/export in different ways depending on whether or not DFSMSrmm stacked volume support is enabled. When you enable stacked volume support, DFSMSrmm tracks logical volumes and stacked volumes.

When you do not enable stacked volume support, DFSMSrmm does not track the stacked volumes. DFSMSrmm supports the IBM Peer-to-Peer VTS by ensuring that you cannot use the names of distributed VTS libraries with DFSMSrmm. You must only use the names of consolidated libraries with DFSMSrmm.

Defining stacked volumes to DFSMSrmm

You can define stacked volumes whether stacked volume support is enabled or not. DFSMSrmm defines the stacked volumes automatically at export time when stacked volume support is enabled. There is no automatic host notification when stacked volumes enter the VTS, leave the VTS, or change category, so you must use DFSMSrmm commands if you want to have all stacked volumes defined to DFSMSrmm.

Recommendation: Always define stacked volumes to DFSMSrmm. This ensures that the volumes cannot be used outside of the VTS. It also ensures that DFSMSrmm checks at cartridge entry processing time that a physical volume or logical volume does not duplicate a stacked volume.

1.1.9 Catalog synchronization

When you run DFSMSrmm with user catalogs and the DFSMSrmm control data set unsynchronized, DFSMSrmm issues catalog locates as required to check if data sets are cataloged. Catalog locates use the standard catalog search to find if a data set is cataloged. Any data sets cataloged using JOBCAT or STEPCAT might not be found to be cataloged, because the standard catalog search might not find them.

When you run DFSMSrmm with user catalogs and the DFSMSrmm control data set synchronized, DFSMSrmm does not need to issue catalog locates to find if a data set is cataloged. The catalog status tracked by DFSMSrmm in the control data set is used to determine if a data set is cataloged. When DFSMSrmm tracks catalog status, it does so regardless of whether JOBCAT or STEPCAT is used. During catalog synchronization, DFSMSrmm uses the Catalog Search Interface (CSI) to retrieve data set catalog information. CSI returns catalog information for all data sets in all catalogs that are in or connected to the master catalog. Because of this, DFSMSrmm can detect a data set is cataloged even if it cannot be found using the standard catalog search.

If you have switched on catalog synchronization, you do not need shared ICF-user catalogs to retain tape data sets as long as they are cataloged. If you have unshared ICF-user catalogs you have to split your daily inventory management. This is because you have to run the expiration processing on each system that has unshared ICF-user catalogs to return expired volumes back to scratch status. In Figure 1-28 the ICF-user catalogs are shared between systems SYS1 and SYS2, but not with SYS3, so you have to run the expiration processing on either system SYS1 or SYS2, and on system SYS3. All other functions can run on any system sharing the DFSMSrmm control data set.

Refer to “Catalog synchronization” on page 329 and the manual *DFSMSrmm Implementation and Customization Guide*, SC26-7405, chapter *“Maintaining the Control Data Set”* for more details.

Archive

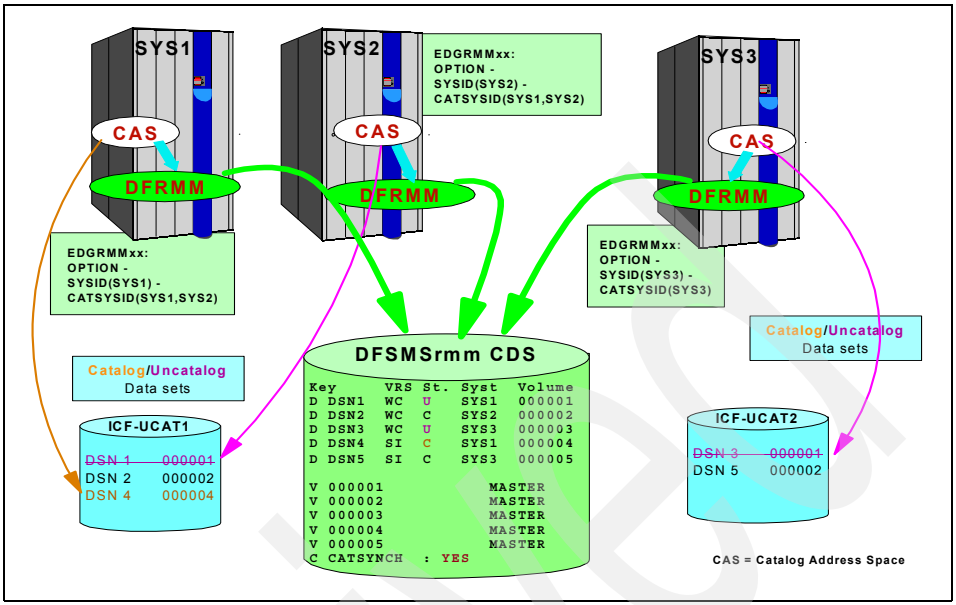


Figure 1-28 Catalog status tracking

1.2 Conversion process overview

The conversion process we describe in this book transfers your tape data information from your current tape management system database to the DFSMSrmm database, while continuing to meet your tape management requirements. Figure 1-29 shows how detailed data about tape volumes is converted to DFSMSrmm format.

Archive

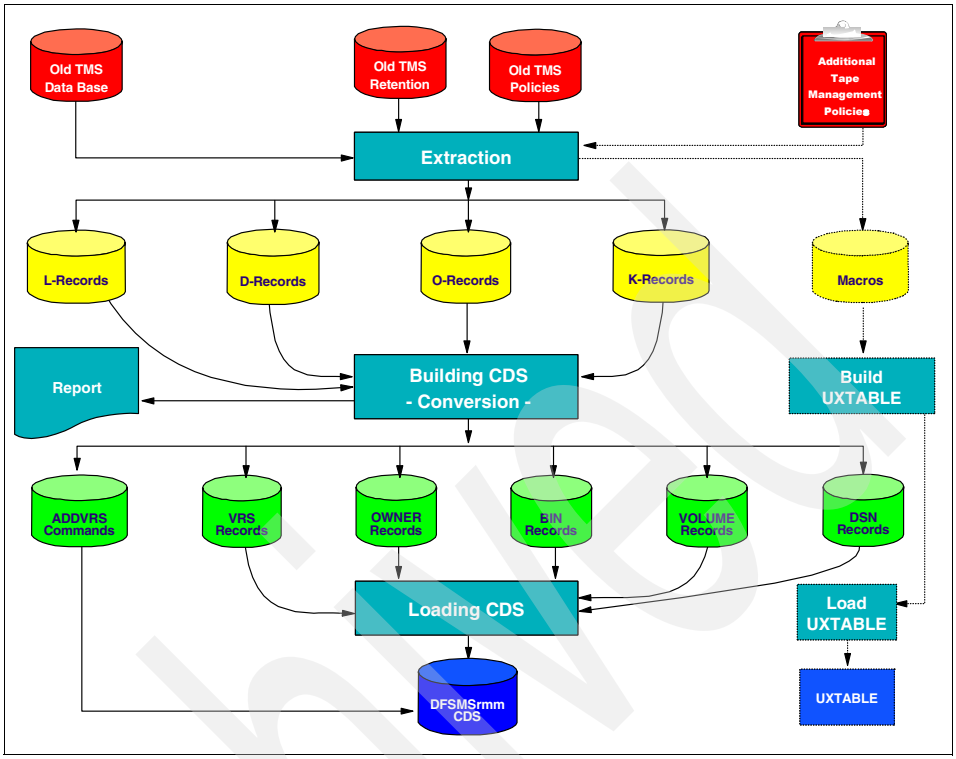


Figure 1-29 Conversion process overview

The data transfer is really a three-phase method using programs and tools shipped with the DFSMSrmm product. You use the programs and tools described in “Chapter 6, “Data extraction” on page 205 to extract the required data that is used in the conversion process.

- The first phase is the data extraction phase, where the data that DFSMSrmm needs is derived from your TLMS data.
- The second phase, the convert phase, is the manipulation of the extracted data into a format that is, when loaded into a VSAM KSDS, useful to DFSMSrmm. The EDGCNVT and EDGCVOVL programs accomplish the second phase.
- The third phase, referred to as the postprocessing phase, is the loading of the VSAM KSDS, the addition of the DFSMSrmm CDS control record, and the utility run that cross checks the records in the DFSMSrmm database, one against the other, to verify completeness and accuracy.

Archive

The conversion process involves more than simple data transfer. It includes testing required functions, validating requirements, and educating users on the changes involved. In this book we describe the three phases of transferring data and help you plan for execution of the complete conversion process.

You can convert to DFSMSrmm in several different ways. For example, you could install DFSMSrmm to manage a set of volumes that your TLMS system does not manage. Over time you can increase the number of volumes that DFSMSrmm manages and reduce the number of volumes that your existing system manages.

Another way of converting to DFSMSrmm — the conversion process that we recommend, and that the rest of this book describes — is to convert all existing volumes from your TLMS management to DFSMSrmm. You convert your current management requirements into DFSMSrmm policies, and, after a period during which you validate that those requirements are being met, you remove your existing TLMS management.

Included in the conversion process is the requirement to expand or consolidate the number of systems and tape databases. For example, you may have two tape databases on two systems today, but you may want to convert to a single database that is shared by the two systems or used by a single image produced from merging the two systems.

The process consists of the following stages:

1. **Learning about DFSMSrmm**
   In stage 1 you ensure that you have enough knowledge about DFSMSrmm to prepare for and plan the conversion.
2. **Planing for conversion**
   In stage 2 you use your knowledge of your current TLMS management and the conversion process to build a plan.
3. **Starting DFSMSrmm**
   In stage 3 you implement DFSMSrmm on your system. You install and test basic DFSMSrmm functions without impacting your current tape management.
4. **Analyzing current environment**
   In stage 4 you analyze your current tape management policies to understand which data to convert.
5. **Extracting data for conversion**
   In stage 5 you use the supplied data extract tools to pull details of tape data from your TLMS management.
6. **Building the DFSMSrmm CDS**
   In stage 6 you convert the data extracted from your TLMS to DFSMSrmm format, and load it into a DFSMSrmm CDS ready for use.

Archive

7. **Parallel running and validation**
   In stage 7 you run DFSMSrmm in parallel with your TLMS management, and ensure that DFSMSrmm provides the management functions you require.
8. **Cutover to production**
   In stage 8 you disable your TLMS management and make DFSMSrmm control all the tape activity.
9. **Measuring success**
   In stage 9 you check that the DFSMSrmm management is what you planned, and take the needed actions for correcting the deviations.
10. **Full exploitation**
    In stage 10 you take advantage of some DFSMSrmm functions that will help improve the management of removable media in your installation.
11. **Uninstall all vendor tape management code from your system**
    In stage 11 you will uninstall any vendor tape management code from your system to clean up your system and DFSMSrmm is your only tape management system.

The structure of this book is geared to the conversion process that you should follow.

Archived

Archived

2

Planning for conversion

In this chapter we introduce you to the conversion process and help you prepare a plan for conversion to DFSMSrmm.

Objectives of this chapter:

- Outline the key steps in converting to DFSMSrmm.
- Help formulate a plan and schedule.
- Identify the milestones and objectives.
- Introduce the stages that are covered in subsequent chapters.
- Identify potential impacts on the production environment, such as IPLs, and the deactivation and removal of TLMS.

At the end of this chapter you will understand the process to follow for conversion to DFSMSrmm, and be able to develop a plan to execute that process.

Archived

© Copyright IBM Corp. 2004. All rights reserved.

2.1 Planning considerations

As you prepare for conversion, there are many factors to consider, such as:

- Software maintenance levels
- Timing of events in your installation
- Change management
- Limiting business risk

2.1.1 Software maintenance levels

You should know the exact release levels of TLMS and the proposed level of DFSMSrmm. The level of DFSMSrmm can alter the functions that are available. The higher the maintenance level, the more new functions are available for you to use. DFSMSrmm ships some new functions through PTFs, so we recommend that you make sure the level to which you plan to convert is the highest available.

Once you have started down the conversion path and have performed testing, we recommend that you install only the maintenance required to fix problems you have experienced. This way you avoid having to repeat tests just to validate that a new software level has not regressed tests you have already completed. Once DFSMSrmm is in production you can revert to your usual software maintenance strategy.

2.1.2 Timing of events in your installation

Be aware of the key events and changes planned for your installation over the potential conversion period. It is possible that any of these events could have an impact on your plan, or that changes you plan to make could impact events that are already in plan.

2.1.3 Change management

As you develop your plan, use your existing change management process to add events to your installation's change schedule. This approach will ensure that any conflicts in what you plan can be resolved as early as possible and enable you to identify where impacts may arise if the plan does not execute as expected.

Archived

2.1.4 Limiting business risk

A key objective of the conversion process is to minimize or eliminate any risk to your business that could occur through the introduction of change into your systems. At each stage we identify what you can do to prepare and test each change so that there are no surprises as DFSMSrmm is introduced. Because you can run DFSMSrmm in parallel with TLMS, you do not need to make a commitment to DFSMSrmm before you are actually ready to commit. You can take time to ensure that DFSMSrmm is meeting your business needs.

If you follow the conversion process described in this book, and make decisions appropriate for your business, you will minimize your business risks.

2.2 Conversion process

The objective of conversion is to use DFSMSrmm rather than AutoMedia or ASG-ZARA to meet your tape management requirements. You can convert to DFSMSrmm in different ways. For example, you can:

- Define your volumes to DFSMSrmm by using the DFSMSrmm ADDVOLUME subcommand or the DFSMSrmm API.
- Use DFSMSrmm to manage a set of volumes that are not managed by AutoMedia or ASG-ZARA. Gradually increase the number of DFSMSrmm-managed volumes and remove volumes from AutoMedia or ASG-ZARA.
- Define all known volumes to DFSMSrmm and run DFSMSrmm in parallel with AutoMedia or ASG-ZARA until DFSMSrmm knows all details about the volumes.
- Use the conversion program and sample extract programs provided with DFSMSrmm to directly convert your AutoMedia or ASG-ZARA data and policies to DFSMSrmm format.

  Once data is converted, you should disable and remove AutoMedia or ASG-ZARA.
- Analyze your AutoMedia or ASG-ZARA setup and review your tape management policies for current applicability. Build a customized environment for DFSMSrmm that meets your installation's needs. Use the conversion program and sample extract programs provided with DFSMSrmm to directly convert your AutoMedia or ASG-ZARA tape management data and policies to a DFSMSrmm format. Run DFSMSrmm in parallel for a period of time to validate that the conversion is successful and that DFSMSrmm continues to meet your needs.

You can use any of the above processes to convert to DFSMSrmm, but only one process, the last process, includes the activities required to prepare for and validate conversion. This is the process that we recommend. In the rest of this book, we describe it in detail.

If you plan to use one of the other processes, you will still find that the information we discuss and present is useful. Therefore, we suggest that you review and use the information and learn from the experiences and knowledge that we draw on to put the process together.

DFSMSrmm provides functions that help with the conversion process. For example, you can use the DFSMSrmm running modes — manual, record, warning, and protect — to help you start DFSMSrmm on the same system on which you run AutoMedia or ASG-ZARA. You can start DFSMSrmm in manual mode and use the DFSMSrmm commands and utilities without DFSMSrmm recording tape activity. Any mode other than manual activates the DFSMSrmm tape recording function.

By using the different modes, you can use DFSMSrmm to gain experience and to educate those that are involved in the conversion.

Note: Before starting DFSMSrmm, ensure that you have read and understand the information provided in Chapter 3., “Preparing the environment” on page 59. If you start DFSMSrmm with unsuitable options, you can potentially have an impact on your AutoMedia or ASG-ZARA environment.

As we describe the conversion process and take you through it, we identify where you should perform testing and what type of testing to perform. You do not test the DFSMSrmm product, but test your use of the product to ensure that DFSMSrmm can meet your needs. Testing is performed and described in:

- Chapter 3, “Preparing the environment” on page 59
- Chapter 6, “Data extraction” on page 205
- Chapter 7, “Building the CDS” on page 263
- Chapter 9, “Cutover to production” on page 357

Once you have converted to DFSMSrmm, take advantage of the functions provided in the DFSMSrmm product.

Archive

2.3 Conversion stages

In this redbook we present information sequentially, in the order of the stages in the conversion process. However, your route through the process probably will not follow the sequential stages exactly. At times you will have to repeat a stage or step after modifying DFSMSrmm or the data. Such repetition will help you get as close as possible to the DFSMSrmm setup that will meet your requirements.

Note: Ensure that your release of DFSMSrmm has the latest PTFs applied.

The conversion process flow is diagrammed in Figure 2-1.

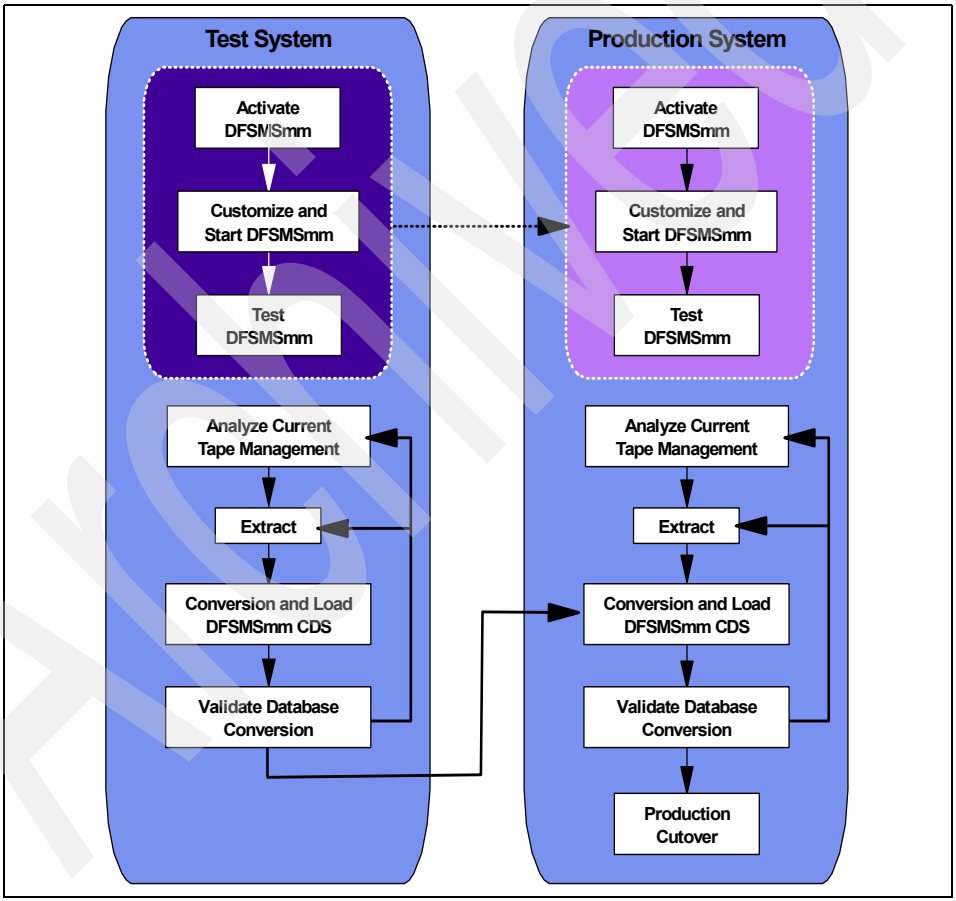


Figure 2-1 Conversion process flow

Archive

You can see that you must follow the same basic process in both test and production environments. In this book we guide you through the conversion process in a test environment. When you actually perform the conversion in production, you will be able to execute the steps easily with minimum risk.

If you have multiple MVS images or a shared disk complex, you must repeat several of the conversion stages for each image or complex. We recommend which stages to repeat at the time we describe the stage.

The conversion stages are:

- Preparing a conversion plan
- Learning about DFSMSrmm
- Starting DFSMSrmm
- Analyzing the current environment
- Extracting data for conversion
- Completing conversion
- Parallel running and validation
- Cutover to production
- Measuring success
- Full exploitation
- Uninstall all vendor tape management code from your system

Each stage has a reference to the chapter that covers it in detail.

2.3.1 Preparing a conversion plan

You can increase the chances of a successful conversion by managing the conversion as a project. This approach ensures that the required tasks are performed, and that the activities are tracked and managed.

It is likely that many other activities will be occurring in your installation at the same time. With so much going on, it is all too easy to forget about a task or assume that it has been completed. Although managing the conversion as a project may appear to be unnecessary, changing AutoMedia or ASG-ZARA will affect many different groups of people in your installation, and your project will be able to keep these groups apprised of the conversion activities.

To ensure a successful conversion, we recommend that you:

- Appoint a project manager
- Appoint a project leader
- Set objectives
- Create a plan and schedule
- Customize and use the checklist provided in Appendix A, “Conversion checklist” on page 607 to track tasks and measure success.

Archived

Use this book to help build a plan for conversion. Each chapter identifies tasks that are recommended and tasks that are mandatory.

As you customize the checklist, you will begin to understand which activities are required and how they must be managed in your installation. Based on this information, you can construct a schedule and add the key steps to your change management system.

Some stages may require that you do an IPL or stop using tapes for short periods of time. These activities may have an impact on your production environment, so you should schedule them around other activities.

We recommend that you test the success of your DFSMSrmm installation, your planned use of DFSMSrmm, and any customization of system options, parameters, and exits. You will need a test system, or you will have to set up a test environment where you can run DFSMSrmm without impacting your production systems.

The sample schedule shown in Figure 2-2 is representative of a typical conversion project. Within the conversion plan there are specific points at which data is converted, and an allowance has been made for a period of parallel running.

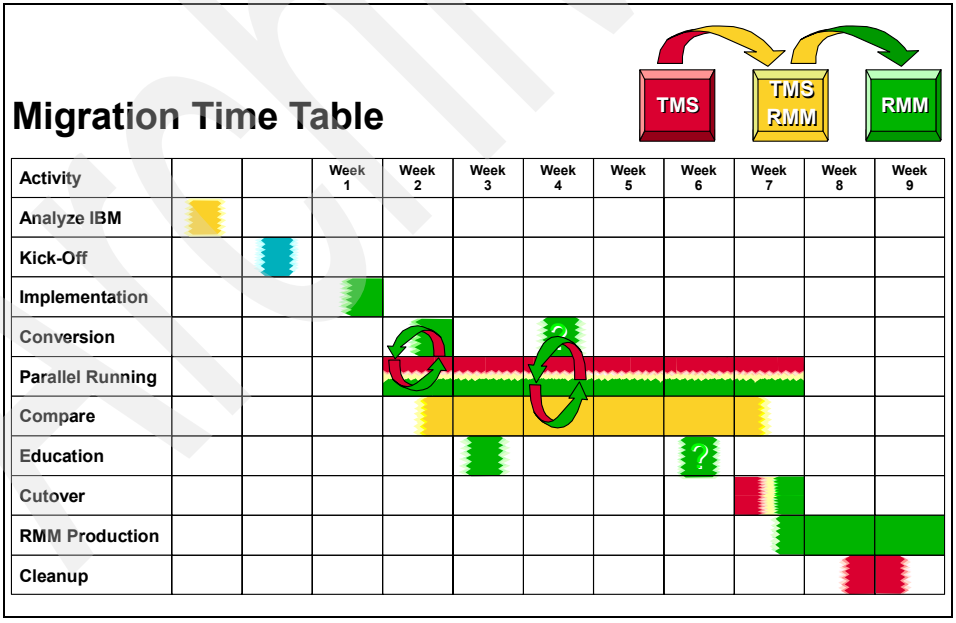


Figure 2-2 Sample schedule

We do not want to dictate how long it should take to execute a conversion plan. We have experience with schedules that are much shorter and much longer than indicated in the above schedule, but the sample schedule is realistic and allows sufficient time for learning a new product and gaining experience with its use. Use the schedule as a model for your own schedule and as input to your existing change management and activity schedules.

Once you have customized the checklist in Appendix A, “Conversion checklist” on page 607, and have prepared a schedule, you are ready to proceed with the conversion process. Establish checkpoints along the way and review progress with the implementation team and your management.

2.3.2 Learning about DFSMSrmm

Before converting data to DFSMSrmm you must have some knowledge of the DFSMSrmm product. Chapter 1, “Introduction” on page 1 provides a quick introduction to DFSMSrmm. You can also gather more information by reading the redbook *DFSMSrmm Primer,* SG24-5983.

You can use the DFSMSrmm product itself to help you learn about DFSMSrmm. Use the information in Chapter 3., “Preparing the environment” on page 59 to get DFSMSrmm started in a test environment, and then use the DFSMSrmm commands and ISPF dialog to gain experience. Ensure that the volumes you use are not the volumes that AutoMedia or ASG-ZARA requires or manages.

All system users and support people (for example, administrators, librarians, system programmers, and operators) who have any tape management responsibilities or use tape volumes require some education and knowledge about DFSMSrmm. Use the DFSMSrmm knowledge and experience that you gain to demonstrate the product to your users and colleagues. All other users should not see any changes in the way their tape data sets are managed before or after the conversion. This means that no JCL changes are required.

2.3.3 Starting DFSMSrmm

At this stage you execute the steps necessary to get the DFSMSrmm product installed and started on one of your systems. This stage includes basic product setup and verifying that DFSMSrmm is installed correctly. See Chapter 3, “Preparing the environment” on page 59 for a detailed explanation of this stage.

If you are using the DFSMS product, you already have the DFSMSrmm component installed. Some of the DFSMSrmm product is in the system link pack area (LPA), so you must IPL your system to ensure that the product code is installed into the LPA. Maintenance to DFSMSrmm may also update the DFSMSrmm code that resides in the LPA. You should plan an IPL before starting DFSMSrmm after the initial SMP/E install, or after applying maintenance to DFSMSrmm code that resides in LPA.

2.3.4 Analyzing the current environment

Before you can proceed with the conversion, you must understand which tape management functions and policies you are using today. You gain this understanding by analyzing your AutoMedia or ASG-ZARA implementation.

Analysis covers:

1. Inventory data gathering
   - Volumes
   - Media
   - Policies
     - Movement
     - Retention
   - Locations
   - Reports
2. Functions used
3. Interfaces to other products

You must extract the data from your AutoMedia or ASG-ZARA database using programs and tools shipped with the DFSMSrmm product. Chapter 6, “Data extraction” on page 205 explains the activities needed for an indepth analysis.

2.3.5 Extracting data for conversion

Before you can complete the conversion, you must extract the data from your AutoMedia or ASG-ZARA database using programs and tools shipped with the DFSMSrmm product.

Chapter 6, “Data extraction” on page 205 covers the data extraction process.

Archived

2.3.6 Completing conversion

During this stage you build a DFSMSrmm CDS using the volume and data set details and the policy details extracted from AutoMedia or ASG-ZARA. Typically, you would convert one database to a single DFSMSrmm CDS. However, you may have a requirement to consolidate systems or databases, and may want to convert the consolidated database to an existing DFSMSrmm CDS.

Chapter 7, “Building the CDS” on page 263 explains what you must do to complete the conversion.

2.3.7 Parallel running and validation

This is a key stage in the conversion process. You use it to ensure that the data converted to DFSMSrmm is correct, and that the converted policies provide the same retention that AutoMedia or ASG-ZARA provides.

Chapter 8, “Parallel running and validation” on page 309 provides detailed information on preparing DFSMSrmm and the system for parallel running. During this stage you may need to repeat the data conversion, make minor adjustments to the policies used with DFSMSrmm, or change the control statements for running the sample extract programs or the EDGCNVT convert program.

During this stage it is important that you minimize the manual changes made to AutoMedia or ASG-ZARA to avoid discrepancies with DFSMSrmm. The conversion process is easily repeated, so you can repeat it whenever necessary. Each time you repeat the data conversion the tape management systems should be inactive to ensure that both products have the same point-in-time information.

The parallel running and validation stage can be as long or as short as you want it to be. For example, you may want to run in parallel for a month so that the different retention policies used during the month can be validated.

Once you have confirmed that the conversion is successful, and validation confirms that DFSMSrmm is providing the functions you require, you can proceed to the next stage.

2.3.8 Cutover to production

Chapter 9, “Cutover to production” on page 357 provides details of how to switch from AutoMedia or ASG-ZARA to DFSMSrmm. As part of the cutover, you may need to change system exits, startup options, and inventory management. Also, you must now disable or remove AutoMedia or ASG-ZARA. You may need to do an IPL to implement some of the changes.

We recommend that you have a fallback or recovery plan in place in case any of the cutover actions cause problems.

2.3.9 Measuring success

Appendix A, “Conversion checklist” on page 607 provides a sample checklist for you to use to track the conversion tasks and verify that production cutover is successful. After you have prepared your plan, ensure that any additional key tasks are identified and added to the checklist. For each task include target dates and who is responsible so that you can manage and measure your plan.

2.3.10 Full exploitation

Once the conversion is successful you should consider implementing additional DFSMSrmm functions. We recommend that you consider doing this soon after cutover to production, as your knowledge of DFSMSrmm is fresh, and you may not have an opportunity to come back to add these functions because of other activities in your installation.

Some of the additional functions that you can implement are detailed in 9.11.3, “Implementing additional functions” on page 388.

2.3.11 Uninstall all vendor tape management code from your system

After your successful DFSMSrmm cutover to production, you may consider to uninstall and delete AutoMedia or ASG-ZARA completely.

Archive

Archived

3

Preparing the environment

In this chapter we describe how to start DFSMSrmm on your system and test DFSMSrmm functions without affecting your TLMS environment.

Objectives of this chapter:

- Implement and start DFSMSrmm on a test system or in a test environment.
- Provide a base for learning about DFSMSrmm.

At the end of this chapter you will have DFSMSrmm started in a test system and will have set it up for basic operations in your environment. You will be ready to create a CDS from your current TLMS system and start customization.

Archived

© Copyright IBM Corp. 2004. All rights reserved.

3.1 DFSMSrmm implementation

In this section we guide you through the process of getting DFSMSrmm started on your system with minimal effort and without impacting any existing tape management system you might have. The intention is to do only essential customization and setup in order to get DFSMSrmm started quickly. In a later stage, when you have more knowledge of DFSMSrmm and your TLMS environment is ready for conversion, you will tailor and customize the DFSMSrmm installation to more specifically meet your needs.

It is essential that you introduce DFSMSrmm in your installation in a way that avoids any impact to your current processing. It is possible to do that, but you must select the correct DFSMSrmm PARMLIB options and ensure that the correct installation-wide exits are in use. When you start DFSMSrmm, TLMS can continue to run and be unaffected.

Although we recommend using test data and test volumes initially with DFSMSrmm, you can start DFSMSrmm by using a CDS that you create during the extract and conversion steps. If you start DFSMSrmm with a converted CDS, you should consider the environment within which you plan to run or test. Consider whether the test environment shares any data, catalogs, or volumes with your production environment. Any changes that DFSMSrmm makes as a result of processing can possibly update shared data. Use the correct DFSMSrmm PARMLIB options to prevent updates to RACF, the TCDB, and catalogs.

No matter how you plan to use DFSMSrmm, when you introduce it into any of your systems, it becomes visible to operators and perhaps some users. Those who will come into contact with DFSMSrmm should be provided with at least basic information about the product and how it affects them.

Follow the steps to implement DFSMSrmm as documented in this redbook, and refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405 for more details.

3.1.1 Installation wizards

Our wizards are interactive assistants that ask you a series of questions about the task you want to perform (for example, setting up DFSMSrmm). The wizards simplify your planning and configuration needs by exploiting recommended values and by building customized checklists for you to use. For configuration tasks, our wizards also generate outputs like jobs, policies, or PARMLIB members that you can upload to z/OS and use.

You can find the DFSMSrmm migration wizard in the following Internet addresses, according to your installed operating system:

```
http://www.ibm.com/servers/eserver/zseries/zos/wizards/
http://www.ibm.com/servers/s390/os390/wizards/
```

These interactive versions of the *DFSMSrmm Implementation and Customization Guide*, SC26-7405 publication will help you find information that you can use to plan your implementation of the latest DFSMSrmm functions and features.

3.1.2 Update installation-wide exits

The installation-wide exits that you install with DFSMSrmm replace any previously installed exits. The DFSMSrmm supplied programming interface and exits are:

- DFSMShsm tape volume programming interface
  - **EDGTVEXT**
- DFSMSdfp MSGDISP exit for updating tape drive displays
  - **IGXMSGEX**
- OAM exits:
  - **CBRUXENT** (cartridge entry exit)
  - **CBRUXEJC** (cartridge eject exit)
  - **CBRUXCUA** (cartridge change use attributes exit)
  - **CBRUXVNL** (volume not in library exit)

The EDGDFHSM programming interface is used by DFSMShsm and Tivoli Storage Manager (TSM) to tell DFSMSrmm which tapes are no longer in use by these products. The equivalent interface between DFSMShsm and TLMS is provided by the ARCTVEXT exit shipped with this product.

Before you install DFSMSrmm, you must determine if any of the DFSMSrmm supplied exits are used on your system. If they are used, and you still require the functions after conversion or during parallel running to DFSMSrmm, you must merge your existing exits with the DFSMSrmm supplied exits.

If you will share one or more exits between TLMS and DFSMSrmm, you must check if updates are needed in any of the exits to achieve coexistence when running in parallel.

There are two ways to do this. You can merge your existing exits or use a router exit which invokes your existing exit and the DFSMSrmm exit. Samples of the ARCTVEXT, the OAM CBRUXENT and CBRUXEJC, and the message IGXMSGEX exits are provided with DFSMSrmm. A sample router exit for ARCTVEXT, and modified samples for IGXMSGEX, CBRUXENT, and CBRUXEJC for parallel running are provided with this redbook as additional

Archived

material. The ARCTVEXT router exit is a front-end to call **TMSTVEXT** (your current renamed ARCTVEXT) and **RMMTVEXT** (the renamed EDGTVEXT shipped with DFSMSrmm).

Note: Create one front-end exit for each pair of exits, if you want both DFSMSrmm and TLMS to be called.

3.1.3 Update and validate PARMLIB members

Update the following SYS1.PARMLIB members:

IEFSSNxx	Subsystem definition
IKJTSOxx	TSO/E commands and programs
IFAPRDxx	Product enablement policy
EDGRMMxx	DFSMSrmm options
SMFPRMxx (optional)	System management facilities (SMF) parameters
GRSRNLxx (optional)	Global resource serialization resourcename lists

Perform this step once for each MVS image. The details about these members are described in the *MVS Initialization and Tuning Reference,* SA22-7592, and for the use of the MVS commands refer to the *MVS System Commands Summary,* SA22-7628.

IEFSSNxx - Subsystem definition

There are considerations when defining the DFSMSrmm subsystem name and started procedure name. You should review the information provided in this book before updating the IEFSSNxx PARMLIB member.

You need to make two changes in IEFSSNxx:

- Define a DFSMSrmm subsystem name to MVS.
- Add the name of the subsystem interface initialization program, EDGSSSI, to fully enable DFSMSrmm.

We recommend that you implement the changes in two steps. Define the DFSMSrmm system at this time (see Figure 3-1) and then you can add the name of the subsystem interface later. If you make only the first change at this time, you can choose whether to start the DFSMSrmm procedure and what time best suits your particular situation. If you implement both changes at once, DFSMSrmm rejects all tape mounts until the subsystem procedure is active, or you use the EDGRESET utility to disable the interface.

```
SUBSYS SUBNAME(SMS)          /* SMS                              */
  INITRTN(IGDSSIIN)
SUBSYS SUBNAME(JES2)         /* JES2 PRIMARY SUBSYSTEM START     */
  PRIMARY(YES) START(YES)
SUBSYS SUBNAME(DFRM)         /* Name of the DFSMSrmm subsystem   */
SUBSYS SUBNAME(AOPA)         /* Netview                          */
```

Figure 3-1 Defining DFSMSrmm subsystem name

If you plan to run DFSMSrmm in parallel with any other tape management system, we recommend that you schedule an IPL to set the DFSMSrmm subsystem enabled in IEFSSNxx (see Figure 3-2) prior to the migration and parallel running.

```
SUBSYS SUBNAME(SMS)          /* SMS                              */
  INITRTN(IGDSSIIN)
SUBSYS SUBNAME(JES2)         /* JES2 PRIMARY SUBSYSTEM START     */
  PRIMARY(YES) START(YES)
SUBSYS SUBNAME(DFRM)         /* Name of the DFSMSrmm subsystem   */
   INITRTN(EDGSSSI)          /*      activate subsystem          */
SUBSYS SUBNAME(AOPA)         /* Netview                          */
```

Figure 3-2 Defining DFSMSrmm subsystem name with interface enabled

You can use the MVS system command SETSSI to define the DFRM subsystem dynamically. You can issue the SETSSI command from one of the following:

- A console that has master authority
- A console to which an operator with sufficient RACF authority has logged on

To define the DFSMSrmm subsystem, use the MVS system command shown in Figure 3-3.

```
SETSSI ADD,SUBNAME=DFRM
```

Figure 3-3 MVS command SETSSI

In this command:

ADD Specifies that a subsystem is to be dynamically added

SUBNAME Specifies the subsystem name to be dynamically added

You can define the DFSMSrmm subsystem and activate the subsystem interface dynamically using the MVS system commands shown in Figure 3-4.

```
SETSSI ADD,SUBNAME=DFRM,INITRTN=EDGSSSI
SETSSI ACTIVATE,SUBNAME=DFRM
```

Figure 3-4 MVS command SETSSI and activate subsystem interface

You can check the successful add of the subsystem by using the MVS system command shown in Figure 3-5.

```
DISPLAY SSI
```

Figure 3-5 MVS command SETSSI

Example 3-1 shows the output of the command.

Example 3-1 DISPLAY SSI command output

```
IEFJ100I  19.31.54  SSI DISPLAY 121
 SUBSYS=JES2 (PRIMARY)
    DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=REJECT
 SUBSYS=MSTR
    DYNAMIC=NO      STATUS=ACTIVE       COMMANDS=N/A
 SUBSYS=SMS
    DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=REJECT
 SUBSYS=J21S
    DYNAMIC=YES     STATUS=INACTIVE     COMMANDS=REJECT
 SUBSYS=J21A
    DYNAMIC=YES     STATUS=INACTIVE     COMMANDS=REJECT
 SUBSYS=RACF
    DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=REJECT
 SUBSYS=OAM1
    DYNAMIC=YES     STATUS=INACTIVE     COMMANDS=REJECT
 SUBSYS=BPO1
    DYNAMIC=YES     STATUS=INACTIVE     COMMANDS=REJECT
 SUBSYS=BPO2
    DYNAMIC=YES     STATUS=INACTIVE     COMMANDS=REJECT
 SUBSYS=DFRM
    DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=REJECT
 SUBSYS=OPCC
    DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=REJECT
 SUBSYS=OPCT
    DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=REJECT
```

To deactivate the DFSMSrmm subsystem DFSMSrmm interface dynamically, use the MVS system command shown in Figure 3-6.

Archived

```
SETSSI DEACTIVATE,SUBNAME=DFRM
```

Figure 3-6 MVS command SETSSI to deactivate subsystem interface

IKJTSOxx - TSO/E commands and programs

Update IKJTSOxx to authorize DFSMSrmm commands and utilities. If you use the TSO CSECT facility rather than IKJTSOxx, refer to *TSO/E Customization,* SA22-7783, for more information.

To authorize the TSO RMM command, add the statement in Figure 3-7.

```
AUTHCMD NAMES(...
              RMM              /* DFSMSrmm TSO commands        */
              ... )
```

Figure 3-7 IKJTSOxx operand AUTHCMD

To authorize the DFSMSrmm utilities that you can use within TSO, or call through the TSO Service Facility, add the following statements shown in Figure 3-8 and Figure 3-9.

```
AUTHTSF NAMES(...
              EDGHSKP          /* DFSMSrmm utility             */
              EDGUTIL          /* DFSMSrmm utility             */
              EDGRPTD          /* DFSMSrmm utility             */
              EDGAUD           /* DFSMSrmm utility             */
              ... )
```

Figure 3-8 IKJTSOxx operand AUTHTSF

```
AUTHPGM NAMES(...
              EDGHSKP          /* DFSMSrmm utility             */
              EDGUTIL          /* DFSMSrmm utility             */
              EDGRPTD          /* DFSMSrmm utility             */
              EDGAUD           /* DFSMSrmm utility             */
              ... )
```

Figure 3-9 IKJTSOxx operand AUTHPGM

To use the updated version of the IKJEFTxx member dynamically, use one of the commands shown in Figure 3-10 in your TSO session or on the MVS console.

Archived

```
TSO PARMLIB UPDATE(xx) or SET IKJTSO=xx
```

Figure 3-10 TSO command PARMLIB update

You can check the successful addition of the RMM command and utilities using the TSO or MVS command shown in Example 3-11.

```
TSO PARMLIB LIST or DISPLAY IKJTSO,ALL
```

Figure 3-11 TSO command PARMLIB list

Example 3-2 shows you the output of the TSO PARMLIB LIST command.

Example 3-2 Output of the TSO PARMLIB list command

```
TSO/E PARMLIB SETTINGS :

  SYS1.PARMLIB(IKJTS000) on volume 037CAT
  Activated by **IPL** on 2003-09-13 at 20:23:00 from system SC65
  Applies to :    SC65
       ...

       CURRENT PARMLIB SETTINGS FOR AUTHCMD

       ADYOPCMD  ADA       EXITEDIT  RACTRACE  RACTR     RECEIVE
       TRANSMIT  XMIT      LISTB     LISTBC    SE        SEND
       PARMLIB   IKJPRMLB  TESTAUTH  TESTA     RMM       MVPXDISP
       LIBLIST   Q         QCBXA     QCBESA    DITTO     DITTOA
       DITTOU    TL        RACONVRT  SYNC      LISTD     LISTDS
       RACONVRT  CONSPROF  AD        ADDSD     AG        ADDGROUP
       AU        ADDUSER   ALG       ALTGROUP  ALD       ALTDSD
       ALU       ALTUSER   BLKUPD    CO        CONNECT   DD
       DELDSD    DG        DELGROUP  DU        DELUSER   LD
       LISTDSD   LG        LISTGRP   LISTDATA  LU        LISTUSER
       RALT      RALTER    RDEF      RDEFINE   RDEL      RDELETE
       RE        REMOVE    RL        RLIST     RACLINK   RVARY
       RACDCERT  PASSWORD  PW        PE        PERMIT    SHCDS
       SETCACHE  SETR      SETROPTS  SINFO     SR        SEARCH
       BACKUP    OPINFO    OPSTAT    SRSTAT    WSSTAT    FREMIGID
       HCEMCS    ARMDREG   CQUERY    CESTPATH  CESTPAIR  CSUSPEND
       CDELPAIR  CDELPATH  CRECOVER  XQUERY    XSTART    XEND
       XSET      XADDPAIR  XDELPAIR  XSUSPEND  XRECOVER  IRRDPI00
       MVPXDISP  NETSTAT   TRACERTE  RMM       CSFTTKE   VLFNOTE
       FCESTABL  FCQUERY   FCWITHDR  MSDPX

       CURRENT PARMLIB SETTINGS FOR AUTHPGM
```

Archived

```
AMASPZAP  CC        EXITEDIT  HESEVMI   ICKDSF    IEBCOPY
IEBGENER  IFCEREP1  Q         QCBXA     QCBESA    CLUPGTRN
CLUPGTPR  CLUPVTPD  CLUPGLOC  DRMAUTH   EDGHSKP   EDGUTIL
EDGRPTD   EDGAUD    HAA$EXCP  ICHDSM00  ICHUT100  ICHUT200
ICHUT400  IDCAMS    IRRDSC00  IRRIRA00  IRRUT100  IRRUT200
IRRUT400  ITPENTER  CSFDAUTH  CSFDPKDS  IXCMIAPU  SINFO
IEARELCN  ARMDREG   IXCMIAPU  IXCM2APU  ADB2UTIL  IRRDPI00


CURRENT PARMLIB SETTINGS FOR AUTHTSF

IEBCOPY   CLUPGTRN  CLUPGTPR  ICQASLIO  ICQASLCO  CLUPVTPD
CLUPGLOC  DRMAUTH   IHVUUSD   ITPENTER  CSFDAUTH  CSFDPKDS
EQQMINOR  EDGHSKP   EDGUTIL   EDGRPTD   EDGAUD    FMNSMF
HAA$EXCP  ADB2UTIL  IKJEFF76

CURRENT PARMLIB SETTINGS FOR NOTBKGND

OPER      OPERATOR  TERM      TERMINAL
```

IFAPRDxx - Product enablement policy

When an installation orders an IBM product such as OS/390 that provides product enablement, IBM supplies a tailored IFAPRD00 member of SYS1.PARMLIB. This tailored member enables the product and any optional features ordered with the product. IFAPRDxx is the PARMLIB member that ensures that you are licensed to use the DFSMS components you want. IFAPRDxx replaced IGDDFPKG in OS/390 Version 1, Release 1.

Review your IFAPRDxx PARMLIB member to verify IFAPRDxx was updated during installation to reflect that DFSMSrmm is licensed for your use (see Figure 3-12). For more information on IFAPRDxx, refer to *z/OS MVS Initialization and Tuning Reference*, SA22-7592.

```
PRODUCT OWNER('IBM CORP')
        NAME(OS/390)
        ID(5647-A01)
        VERSION(*) RELEASE(*) MOD(*)
        FEATURENAME(DFSMSRMM)
        STATE(ENABLED)
```

Figure 3-12 Enabling DFSMSrmm license into IFAPRDxx

You can use the MVS SET PROD operator command (see Figure 3-13) to modify the enablement policy dynamically by specifying which IFAPRDxx member the system is to use. Statements in the IFAPRDxx member modify, not replace, an existing policy. The change to the policy takes place immediately but does not

affect any product instances that are already running. To update the IFAPRDxx member dynamically, use the following MVS system command in Figure 3-13.

```
SET PROD=xx
```

Figure 3-13 MVS command SET

You can use a MVS command as shown in Figure 3-14 to display the active enablement policy.

```
DISPLAY PROD,REGISTERED
```

Figure 3-14 MVS command DISPLAY

Example 3-3 shows you the output of a DISPLAY PROD command.

Example 3-3 Output of a DISPLAY PROD command

```
RESPONSE=SC65
 IFA111I 15.17.53 PROD DISPLAY 213
 S OWNER            NAME             FEATURE          VERSION  ID
 E IBM CORP         z/OS             z/OS             01.05.00 5694-A01
 E IBM CORP         z/OS             BDTFTF           **.**.** 5694-A01
 E IBM CORP         z/OS             BDTNJE           **.**.** 5694-A01
 E IBM CORP         z/OS             DFSMSRMM         **.**.** 5694-A01
 E IBM CORP         z/OS             DFSMSTVS         **.**.** 5694-A01
 E IBM CORP         z/OS             INFOPRINT SERVER **.**.** 5694-A01
 N IBM CORP         z/OS             JES2             01.05.00 5694-A01
 E IBM CORP         z/OS             JES3             01.05.00 5694-A01
 N IBM CORP         z/OS             RACF             **.**.** 5694-A01
 E IBM CORP         z/OS             RMF              **.**.** 5694-A01
 E IBM CORP         z/OS             Security Server  **.**.** 5694-A01
 E IBM CORP         z/OS             SDSF             **.**.** 5694-A01
 E IBM CORP         z/OS             TCP/IP BASE      **.**.** 5694-A01
```

EDGRMMxx - DFSMSrmm options

Create an EDGRMMxx PARMLIB member for DFSMSrmm (see Example 3-4 on page 76). The member name is in the form EDGRMMxx, where *xx* is a two-character alphanumeric suffix of your choice. The default is EDGRMM00. A sample EDGRMMxx member is provided in SAMPLIB member EDGIVPPM. You can copy it to PARMLIB, and modify it according to your installation requirements.

In the sections that follow, we describe the most important options to specify in the EDGRMMxx PARMLIB member when DFSMSrmm is running in parallel to TLMS. Use the following information together with the *DFSMSrmm*

Archived

Implementation and Customization Guide, SC26-7405, which describes in detail the options you can use with DFSMSrmm.

OPTION command

This section details the various OPTION commands used within EDGRMMxx. We recommend that you specify the following OPTION subparameters in EDGRMMxx:

OPMODE(R)

Specifies the mode of DFSMSrmm.

R Record-only mode
You can use DFSMSrmm TSO commands, ISPF dialog, and inventory management functions. In addition, DFSMSrmm records information about magnetic tape volumes used on the system, including details about volume owners and data set names.

For volumes in an automated tape library dataserver, DFSMSrmm records information about entry and eject activities. During entry processing, DFSMSrmm provides volume status information to OAM. If a volume is not already defined to DFSMSrmm, DFSMSrmm automatically defines it as long as there are no conflicts with either the VOLSER or corresponding rack number.

DFSMSrmm does not validate or reject volumes when in record-only mode. When DFSMSrmm is running in record-only mode, the volume status in the TCDB is not updated. When you issue DFSMSrmm commands, the TCDB is updated if appropriate and the volume is system-managed. When volumes are returned to scratch status, no TCDB updates are made.

BACKUPPROC(EDGBKUP)

Specifies the name of the procedure that you want to start automatically when the journal percentage full threshold is reached.

Specify a valid alphanumeric procedure name from 1 to 8 characters. If no name is specified, then no automatic start command is issued.

COMMANDAUTH(OWNER)

Specifies the type of authorization that DFSMSrmm is to check. Specify OWNER when you expect the owners of volume information and data set information to be able to update their own data sets and volumes using RMM TSO subcommands. Specify DSN when you expect changes to volume and data set information to be authorized using the RACF DATASET class and TAPEVOL class.

Archived

DATEFORM(J)

Specifies the date format for messages and reports (use the date format preferred in your computing center).

Value	Language	Format	Example
A	American	mm/dd/yyyy	12/15/2003
E	European	dd/mm/yyyy	15/12/2003
I	ISO	yyyy/mm/dd	2003/12/15
J	Julian	yyyy/ddd	2003/349

DISPDDNAME(LOANDD)

Specifies the name of the DD card that identifies the data set, which contains disposition control statements that DFSMSrmm processes during CLOSE or EOV processing. For detailed information refer to 11.5, “Disposition processing” on page 464.

DISPMSGID(EDG4054I)

Specifies the message number that DFSMSrmm uses for write-to-operator messages specified in the disposition control file.

DSNAME(RMM.PROD.CDS)

Specifies the name of the DFSMSrmm control data set (CDS). Specify a name up to 44 characters long.

Note: If you do not specify DSNAME, you must specify the data set name in the MASTER DD statement in the DFSMSrmm started procedure. If you specify a name for both DSNAME and MASTER DD, DFSMSrmm ignores the MASTER DD statement

JRNLNAME(RMM.PROD.JRNL)

Specifies the name of the journal. Specify a name up to 44 characters long.

Note: If you do not specify JRNLNAME, you can specify a name in the JOURNAL DD statement in the DFSMSrmm started procedure. If you do not specify a journal name in either EDGRMMxx or the started procedure, DFSMSrmm does not provide journaling. If you specify a name in both, DFSMSrmm uses the JRNLNAME value and ignores the JOURNAL DD statement.

Archive

MAXHOLD(100)

Specifies the maximum number of activities DFSMSrmm performs to the CDS before the reserve is released and reacquired. Specify MAXHOLD to minimize the impact that long operations, such as RMM TSO ADD subcommands with large count values, or large searches, have on other users.

MAXRETPD(NOLIMIT)

Specifies the maximum retention period that a user can request for data sets on volumes. Specify NOLIMIT or a value between 0 and 9999 days. When a value between 0 and 9999 days is specified, the value is added to the current date to determine the maximum allowed expiration date. Specify NOLIMIT to use the dates 99365 or 99366, which means the data sets never expire. If the calculated date is 31 December 1999, the expiration date 1 January 2000 is used.

If the DFSMSrmm ISPF dialog or RMM TSO subcommands are used to specify a retention period or expiration date that exceeds the MAXRETPD value, DFSMSrmm fails the subcommand or panel request.

MEDIANAME(3480)

Specify to set a default medianame value that DFSMSrmm uses when you do not specify a media name for a volume. The medianame is used when you add volumes, define pools in your installation, define a default pool for your installation, and when the EDGINERS utility selects volumes for automatic processing.

PREACS(NO)

Specify the PREACS operand to control whether DFSMSrmm-supplied and EDGUX100 installation exit-supplied values are input to SMS pre-ACS processing:

NO Specify NO to avoid DFSMSrmm Pre-ACS processing using the DFSMSrmm EDGUX100 installation exit.

Archived

RETPD(5)

This specifies the default retention period for all new data sets on tape volumes (use a value between 0 and 9999 days). The specified value is added to the current date to determine the expiration date.

Note: The value you specify for the RETPD operand should be the same as the value you use in TLMS.

REUSEBIN(STARTMOVE)

Specify to control how DFSMSrmm reuses bins when when a volume is moving from a bin:

STARTMOVE A bin can be reused as soon as a volume starts moving out of a bin. Extended bin support must be enabled before you can use this operand. For more information, see the *DFSMSrmm Implementation and Customization Guide*, SG26-7405, and the chapter titled “Enabling Extended Bin Support” to enable extended bin support.

Note: The use of STARTMOVE is recommended because most vendor tape management systems will reuse bins directly.

SMSACS(YES)

Specify this operand to control whether DFSMSrmm calls SMS ACS processing to enable use of storage group and management class values with DFSMSrmm:

YES Specify YES to enable DFSMSrmm calls to the SMS ACS processing to obtain management class and storage group names. If values are returned by the SMS ACS routines, the values are used instead of the DFSMSrmm and EDGUX100 decisions.

SMSTAPE(UPDATE(EXITS, SCRATCH, COMMAND), PURGE(YES))

Use SMSTAPE to specify how DFSMSrmm updates the TCDB and controls system-managed tape processing:

UPDATE Use UPDATE to select the system-managed tape functions DFSMSrmm provides. The UPDATE operand has three subparameters: EXITS, SCRATCH, and COMMAND. You can specify one or more of the subparameters. When DFSMSrmm is running in PROTECT mode, DFSMSrmm ignores the UPDATE

operand and performs processing as if you specified EXITS, SCRATCH, and COMMAND. When DFSMSrmm is running in WARNING or RECORD mode, DFSMSrmm does not update TCDB information unless you request the update. You can specify one or more of the values. When you specify a value, DFSMSrmm performs the updates to the TCDB.

EXITS Specify when you want DFSMSrmm volume status information to override the OAM volume status during entry processing and you want to use the DFSMSrmm VNL exit.

SCRATCH Specify when you want DFSMSrmm to update the volume status in the TCDB during expiration processing when volumes are returned to scratch status.

COMMAND Specify when you want to use the RMM TSO subcommands or the DFSMSrmm API to update the TCDB. This controls change of status, TDSI and owner information, eject processing and manual cartridge entry processing.

PURGE Use PURGE to control how DFSMSrmm affects the TCDB volume records during EJECT processing. The default is PURGE(ASIS) in all operating modes except MANUAL mode. In manual mode, DFSMSrmm provides no support for eject processing.

YES Specify when you want DFSMSrmm to force the TCDB volume records to be purged at eject time.

TPRACF(N)

Do not create RACF tape profiles.

Attention: DFSMSrmm will manipulate your RACF TAPEVOL and TAPEDSN profiles in each operating mode if the TPRACF function is set to A or P.

UNCATALOG(N)

Do not use DFSMSrmm uncatalog processing. If UNCATALOG(N) is set, DFSMSrmm does not uncatalog data sets under any circumstances.

Attention: DFSMSrmm will uncatalog data sets in each operating mode if the UNCATALOG function is set to Y or S.

Archived

VRSJOBNAME(2)

Specify this operand to select whether DFSMSrmm uses the job name from VRS as the primary or secondary match.

VRSJOBNAME(2) The data set name takes precedence over the job name during VRS match processing. This is the option that most closely matches the use of job names in TLMS.

VRSEL(NEW)

Use VRSEL to control how DFSMSrmm inventory management vital record processing uses retention and movement information defined in VRSs.

NEW Specify if you want:

- DFSMSrmm to process retention information in NAME VRSs, release options defined in VRSs, and VRSs chained using the RMM ADDVRS operand.
- If you are converting using the conversion tools and require functions close to that given by TLMS.

REJECT command

DFSMSrmm can reject a range of tapes only if you first define them to DFSMSrmm. It can reject all tapes not defined to it, which you specify with a single asterisk for the prefix, but it cannot reject a range of tapes not defined to it.

You can specify multiple REJECT commands to define different ranges of tapes. However, use only one ANYUSE operand and one OUTPUT operand for each REJECT command.

ANYUSE(*)

Specifies the shelf locations of volumes not available for read or write processing. You can specify a generic shelf location consisting of a one to five character prefix followed by an *. Specify a single asterisk if, at OPEN time, you want to reject all volumes not defined to DFSMSrmm. You can use the installation exit, EDGUX100, to request that DFSMSrmm ignore such tapes.

Note: If you specify REJECT ANYUSE(*), all tapes not defined to the DFSMSrmm CDS cause an error message to be issued by DFSMSrmm when the tapes are used.

Archived

SECCLS command

Use the SECCLS command to define security classes for data sets and volumes. These security classes appear in reports and RMM TSO subcommand output. DFSMSrmm records these security classes in the DFSMSrmm CDS only, it does not make RACF aware of them. There is no connection between these definitions and any similar definitions or function provided in RACF, but you can use similar values for overall consistency.

The examples of the SECCLS commands provided in Example 3-4 show the use of the following functions:

- Volumes must be erased on release.
- Issue operator confirmation messages.
- Write SMF security records each time a tape data set in this class is opened.

MNTMSG command

Use the MNTMSG command to tailor mount and fetch messages so they display the VOLSER and pool name or rack number. This information makes it easier for the operator to pull and mount volumes.

VLPOOL command

Use the VLPOOL command to define pools. When you add a new volume to the library, DFSMSrmm assigns it a shelf location from the specified pool. The most important parameters are the following.

TYPE

In DFSMSrmm you can specify two categories of pools:

R A rack pool is shelf space assigned to hold volumes that are generally read-only, and that enter and leave your installation on an ad-hoc basis. These volumes are typically software product volumes and customer volumes and do not adhere to your installation's naming conventions.

S A scratch pool is assigned to shelf space to hold all other volumes in the removable media library. The volumes assigned to this shelf space can be used to satisfy scratch requests as long as the volumes are in scratch status. Once the volume has been written to, it becomes a master volume until the data is no longer required by the installation. The volume remains in the same DFSMSrmm scratch pool in that it occupies the same shelf space regardless of status.

Archived

We recommend that you define all the pools used in TLMS when running in parallel.

RACF(N)

Specifies the type of RACF tape support that DFSMSrmm should provide for volumes in the pool you are defining.

N Specify if you do not want DFSMSrmm to create RACF tape profiles for the volumes in the pool.

When you are defining RACF tape support for volumes in a pool, you must look at the RACF tape support you have defined for your installation with the OPTION TPRACF command. Refer to 9.1.1, “OPTION command” on page 361 for more information about this parameter.

Example 3-4 shows a sample EDGRMMxx PARMLIB member you can use as a model when preparing for DFSMSrmm recording mode.

Example 3-4 Sample EDGRMMxx PARMLIB member

```
OPTION    OPMODE(R)                 /* Record-Only Mode          */ -
          ACCOUNTING(J)             /* Accounting from JOB       */ -
          BACKUPPROC(EDGBKUP)       /* Name of BACKUP-proc       */ -
          BLP(RMM)                  /* DFSMSrmm controls BLP     */ -
          CATRETPD(0012)            /* catalog retention         */ -
          CATSYSID(*)               /* all catalogs shared       */ -
          CDSID(PROD)               /* control data set id       */ -
          COMMANDAUTH(OWNER)        /* type of authorization     */ -
          DATEFORM(J)               /* Date format               */ -
          DISPDDNAME(LOANDD)        /* DISP ctrl  DD card        */ -
          DISPMSGID(EDG4054I)       /* DISP message number       */ -
          DSNAME(RMM.PROD.CDS)      /* CDS data set name         */ -
          IPLDATE(N)                /* IPL date checking         */ -
          JRNLNAME(RMM.PROD.JRNL)   /* JRNL data set name        */ -
          JOURNALFULL(75)           /* Percentage JRNL full      */ -
          LINECOUNT(054)            /* Lines per page            */ -
          MASTEROVERWRITE(LAST)     /* Overwriting of a vol      */ -
          MAXHOLD(100)              /* Number of I/O oper.       */ -
          MAXRETPD(NOLIMIT)         /* Maximum retention         */ -
          MEDIANAME(3480)           /* spec. how to move vols    */ -
          MOVEBY(VOLUME)            /* spec. how to move vols    */ -
          MSG(M)                    /* case for message txt      */ -
          NOTIFY(Y)                 /* Notify volume owners      */ -
          PDA(OFF)                  /* PDA is disabled           */ -
          PDABLKCT(255)             /* number of blocks          */ -
          PDABLKSZ(31)              /* blocksize in K             */ -
          PDALOG(OFF)               /* PDA output disabled       */ -
          PREACS(NO)                /* Disable EDGUX100 ACS pr.  */ -
          RETAINBY(VOLUME)          /* spec. how to retain vols  */ -
```

```
         RETPD(0005)                /* Default retention       */ -
         REUSEBIN(STARTMOVE)        /* reuse BIN as soon as pos.*/ -
         SCRATCHPROC(EDGXPROC)      /* ATL/MTL procedure       */ -
         SMFAUD(248)                /* SMF audit records       */ -
         SMFSEC(249)                /* SMF security records    */ -
         SMSACS(YES)                /* enable MV ACS processing */ -
/*       SMSTAPE(UPDATE(EXITS,SCRATCH,COMMAND),PURGE(YES))    ATL*/ -
         SYSID(EGZB)                /* Name of the system      */ -
         TPRACF(N)                  /* RACF tape support       */ -
         TVEXTPURGE(EXPIRE)         /* set an expiration date  */ -
         UNCATALOG(N)               /* Catalog support         */ -
         VRSCHANGE(INFO)            /* No additional action    */ -
         VRSEL(NEW)                 /* New VRS processing      */ -
         VRSJOBNAME(2)              /* DATASETNAME/JOBNAME     */ -
         VRSMIN(0000000100,WARN)    /* Warn if < 100 VRSs      */
/* ************************************************************** */
MNTMSG   MSGID(IEF233A)      ID(001) VOLUME(016) RACK(016)
MNTMSG   MSGID(IEF233D)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E K')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E R')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E D')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID(IEF455D)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID(IEC501A)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E K')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E D')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E R')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E RD') ID(001) VOLUME(017) RACK(999)
MNTMSG   MSGID('IEC502E RK') ID(001) VOLUME(017) RACK(999)
MNTMSG   MSGID(IAT5110)      ID(001) VOLUME(044) RACK(999)
MNTMSG   MSGID(IAT5210)      ID(001) VOLUME(050) RACK(999)
MNTMSG   MSGID(IAT5410)      ID(001) VOLUME(020) RACK(999)
/* ************************************************************** */
REJECT   ANYUSE(*)
/* ************************************************************** */
VLPOOL   PREFIX(SC*)                                             -
         TYPE(S)                                                 -
         RACF(N)                                                 -
         MEDIANAME(3590)                                         -
         EXPDTCHECK(N)                                           -
         DESCRIPTION('3590 DEFAULT SCRATCH POOL')
VLPOOL   PREFIX(N*)                                              -
         TYPE(R)                                                 -
         RACF(N)                                                 -
         EXPDTCHECK(0)                                           -
         DESCRIPTION('TEST POOL FOR NORBERT')
VLPOOL   PREFIX(F*)                                              -
         TYPE(R)                                                 -
         RACF(Y)                                                 -
         EXPDTCHECK(Y)                                           -
```

```
          DESCRIPTION('FOREIGN CARTRIDGES')
VLPOOL    PREFIX(PP*)                                                  -
          TYPE(R)                                                      -
          RACF(Y)                                                      -
          EXPDTCHECK(Y)                                                -
          DESCRIPTION('SOFTWARE CARTRIDGES')
/* ************************************************************ */
LOCDEF    LOCATION(SAFE)                                               -
          TYPE(STORAGE)                                                -
          PRIORITY(2000)                                               -
          MEDIANAME(*)                                                 -
          MANAGEMENTTYPE(BINS)
LOCDEF    LOCATION(SANJOSE)                                            -
          TYPE(STORAGE)                                                -
          PRIORITY(2000)                                               -
          MEDIANAME(*)                                                 -
          MANAGEMENTTYPE(BINS)
LOCDEF    LOCATION(NEWYORK)                                            -
          TYPE(STORAGE)                                                -
          PRIORITY(2000)                                               -
          MEDIANAME(*)                                                 -
          MANAGEMENTTYPE(NOBINS)
LOCDEF    LOCATION(IBMATL1)                                            -
          TYPE(LIBRARY)                                                -
          PRIORITY(4800)
/* ************************************************************ */
SECCLS    NUMBER(010)                                                  -
          NAME(UCL)                                                    -
          SMF(N)                                                       -
          MESSAGE(N)                                                   -
          ERASE(N)                                                     -
          DESCIPTION('UNCLASSIFIED')                                   -
          MASK('**')
SECCLS    NUMBER(050)                                                  -
          NAME(IUO)                                                    -
          SMF(Y)                                                       -
          MESSAGE(N)                                                   -
          ERASE(N)                                                     -
          DESCIPTION('INTERNAL USE ONLY')                              -
          MASK('**.IUO.**',                                            -
               '**.INTERNAL.**',                                       -
               '**.INT.**')
SECCLS    NUMBER(100)                                                  -
          NAME(IC)                                                     -
          SMF(Y)                                                       -
          MESSAGE(Y)                                                   -
          ERASE(Y)                                                     -
          DESCIPTION('EXTERNAL USE ONLY')                              -
          MASK(**.IC.**)
```

Archived

SMFPRMxx (optional) - System management facilities (SMF) parameters

If you want security or audit records, update SMFPRMxx to define the SMF record numbers that DFSMSrmm generates. In DFSMSrmm, you define the SMF record numbers in the EDGRMMxx PARMLIB member using two operands: OPTION SMFAUD for auditing records and OPTION SMFSEC for security records. For more information on these commands, see the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

Select two SMF record numbers in the 128 to 255 range that your installation is not using. Add your SMFAUD and SMFSEC record numbers to SMFPRMxx as specified in *z/OS MVS System Management Facilities (SMF)*, SA22-7630.

By default, there is no SMF recording.

You can use the MVS command DISPLAY as shown in Figure 3-15 to display data.

```
DISPLAY SMS,O
```

Figure 3-15 Display SMS information

Where:

O Directs the system to display the current SMF options

Example 3-5 shows you the output of the display SMS command.

Example 3-5 Output of a display SMS command

```
RESPONSE=SC65
 IEE967I 16.03.17 SMF PARAMETERS 347
         MEMBER = SMFPRM00
         MULCFUNC -- DEFAULT
         SYNCVAL(00) -- DEFAULT
         DUMPABND(RETRY) -- DEFAULT
         SUBSYS(STC,NOINTERVAL) -- SYS
         SUBSYS(STC,NODETAIL) -- SYS
         SUBSYS(STC,EXITS(IEFUSO)) -- PARMLIB
         SUBSYS(STC,EXITS(IEFUJP)) -- PARMLIB
         SUBSYS(STC,EXITS(IEFUJI)) -- PARMLIB
         SUBSYS(STC,EXITS(IEFACTRT)) -- PARMLIB
         SUBSYS(STC,EXITS(IEFU85)) -- PARMLIB
         SUBSYS(STC,EXITS(IEFU84)) -- PARMLIB
         SUBSYS(STC,EXITS(IEFU83)) -- PARMLIB
         SUBSYS(STC,EXITS(IEFU29)) -- PARMLIB
         SUBSYS(STC,TYPE(0:98,100:255)) -- PARMLIB
```

Archived

```
SYS(NODETAIL) -- PARMLIB
SYS(NOINTERVAL) -- PARMLIB
SYS(EXITS(IEFU29)) -- PARMLIB
SYS(EXITS(IEFUTL)) -- PARMLIB
SYS(EXITS(IEFUJI)) -- PARMLIB
SYS(EXITS(IEFUSO)) -- PARMLIB
SYS(EXITS(IEFUJP)) -- PARMLIB
SYS(EXITS(IEFUSI)) -- PARMLIB
SYS(EXITS(IEFUJV)) -- PARMLIB
SYS(EXITS(IEFACTRT)) -- PARMLIB
SYS(EXITS(IEFU85)) -- PARMLIB
SYS(EXITS(IEFU84)) -- PARMLIB
SYS(EXITS(IEFU83)) -- PARMLIB
SYS(TYPE(0:98,100:255)) -- PARMLIB
NOBUFFS(MSG) -- PARMLIB
LASTDS(MSG) -- PARMLIB
LISTDSN -- PARMLIB
SID(SC65) -- PARMLIB
SID(SC65) -- PARMLIB
DDCONS(NO) -- PARMLIB
JWT(2400) -- PARMLIB
MEMLIMIT(NOLIMIT) -- PARMLIB
STATUS(010000) -- PARMLIB
MAXDORM(3000) -- PARMLIB
INTVAL(10) -- PARMLIB
REC(PERM) -- PARMLIB
NOPROMPT -- PARMLIB
DSNAME(SYS1.SC65.MAN3) -- PARMLIB
DSNAME(SYS1.SC65.MAN2) -- PARMLIB
DSNAME(SYS1.SC65.MAN1) -- PARMLIB
ACTIVE -- PARMLIB
```

GRSRNLxx (optional) - Global resource serialization resourcename list

If you are not using global resource serialization (GRS) to control enqueues among sharing systems, you do not have to update GRSRNLxx. If you have multiple systems connected in a GRSplex, decide how you want GRS to handle the reserve on the CDS:

1. If the volume where the CDS resides does not contain other critical data, and real reserves will not cause any shared disk contention problems, add the reserve name to the GRS exclusion list. This will leave the real hardware reserve in effect and cause the associated SYSTEMS enqueue to be converted to a local SYSTEM enqueue. The delay associated with sending the additional unnecessary enqueue around the GRSplex will then be avoided.

Archived

The required statements in GRSRNLxx to convert the SYSTEMS enqueue to a local SYSTEM enqueue are shown in Figure 3-16.

```
RNLDEF RNL(EXCL) TYPE(SPECIFIC)
QNAME(SYSZRMM)
RNAME(MASTER.RESERVE)
```

Figure 3-16 GRS EXCL definition

2. If the volume where the CDS resides contains other critical data, and real reserves can impact other systems or cause disk lockouts, add the reserve name to the GRS reserve conversion list. This will leave the global SYSTEMS ENQUEUE in effect and remove the real hardware reserve.

   The required statements in GRSRNLxx to convert the RESERVE to a SYSTEMS enqueue are (see Figure 3-17).

```
RNLDEF RNL(CON) TYPE(SPECIFIC)
QNAME(SYSZRMM)
RNAME(MASTER.RESERVE)
```

Figure 3-17 GRS CON definition

Note: If you are not using GRS, and use an equivalent product instead, use the ENQ resource name information provided in the DFSMSrmm publications to define the resource correctly to the other product. Do not use generic or global options for SYSZRMM. You must ensure that each resource with the SYSZRMM major name is processed correctly.

Figure 3-18 shows you the difference between a system enqueue and a hardware reserve.

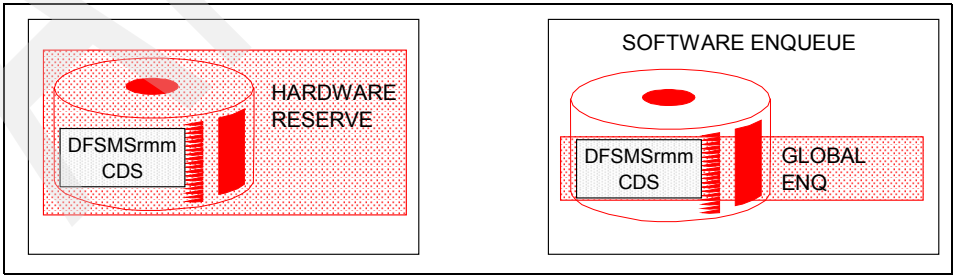


Figure 3-18 Use of GRS

You can use the MVS SET GRSRNL command shown in Figure 3-19 to change the current GRSRNL definitions.

```
SET GRSRNL=(xx[,xx]...)
```

Figure 3-19 Update GRS information

The two alphanumeric characters indicating the GRSRNLxx member(s) of the logical PARMLIB that contain the specified GRS resource name lists. GRSRNL allows you to change the current RNLs specified in one or more GRSRNLxx PARMLIB member. Do not use parentheses when only one PARMLIB member is specified.

Attention: Use extreme caution when issuing the SET GRSRNL command to change heavily used or highly critical resources. Work that requires resources for a critical application, or resources used by the operating system may become suspended or delayed, which can impair the performance of that critical application or the operating system itself.

Note: You cannot specify the GRSRNL= parameter if GRSRNL has already been set to EXCLUDE in the logical PARMLIB member IEASYSxx.

Has sysplex scope only when all systems are sharing the same PARMLIB data set.

You can use the MVS DISPLAY GRS command shown in Figure 3-20 to display your current GRS information.

```
DISPLAY GRS,ALL
```

Figure 3-20 Display GRS information

Example 3-6 contains the result of a DISPLAY GRS command.

Example 3-6 DISPLAY GRS command output

```
RESPONSE=SC65
 ISG343I 16.48.43 GRS STATUS 431
      LIST TYPE QNAME    RNAME
      INCL GEN  SPFEDIT
      INCL GEN  SYSDSN
      INCL GEN  SYSIKJBC
      INCL GEN  SYSIKJUA SYS1
```

```
     INCL SPEC SYSIKJUA VAIN
     INCL GEN  SYSZVOLS
     EXCL SPEC SPFEDIT  PASSWORD
     EXCL SPEC SPFEDIT  SYS1.DCMLIB
     EXCL SPEC SPFEDIT  SYS1.NUCLEUS
     EXCL SPEC SPFEDIT  SYS1.SVCLIB
     EXCL GEN  SYSCTLG
     EXCL SPEC SYSDSN   PASSWORD
     EXCL SPEC SYSDSN   SYS1.DCMLIB
     EXCL GEN  SYSDSN   SYS1.JES3
     EXCL SPEC SYSDSN   SYS1.NUCLEUS
     EXCL SPEC SYSDSN   SYS1.SVCLIB
     EXCL SPEC SYSIGGV2 CATALOG.SHRICF1.VIODFBK
     EXCL SPEC SYSIGGV2 CATALOG.TOTICF1.VO33SMP
     EXCL SPEC SYSIGGV2 CATALOG.TOTICF1.VO33SM1
     EXCL SPEC SYSIGGV2 UCAT.VBOOKO1
     EXCL SPEC SYSIGGV2 UCAT.VOS3R2B
     EXCL GEN  SYSZJES2
     CON  GEN  ARCGPA
     CON  GEN  DSPURIO1
     CON  SPEC HWRESERV BOOKO1
     CON  SPEC HWRESERV BOOKO2
     CON  SPEC HWRESERV BOOKO3
     CON  SPEC HWRESERV BOOKO4
     CON  SPEC HWRESERV IODFBK
     CON  SPEC HWRESERV IODFPK
     CON  SPEC HWRESERV OS3R2B
     CON  SPEC HWRESERV O33SMP
     CON  SPEC HWRESERV O33SM1
     CON  SPEC HWRESERV O33SM2
     CON  SPEC HWRESERV O33SM3
     CON  SPEC HWRESERV O33SM4
     CON  GEN  IGDCDSXS
     CON  GEN  SPFEDIT
     CON  GEN  SPZAPLIB
     CON  GEN  SYSIEWLP
     CON  GEN  SYSIGGV2
     CON  GEN  SYSVTOC
     CON  GEN  SYSZRACF
     CON  SPEC SYSZRMM  MASTER.RESERVE
     CON  GEN  SYSZVVDS
NO ENQ RESOURCE CONTENTION EXISTS
NO REQUESTS PENDING FOR ISGLOCK STRUCTURE
NO LATCH CONTENTION EXISTS
SYSTEM    STATE              SYSTEM    STATE
SC63      CONNECTED          SC64      CONNECTED
SC65      CONNECTED
THERE IS NO RNL CHANGE IN PROGRESS.
GRS STAR MODE INFORMATION
```

Archived

```
LOCK STRUCTURE (ISGLOCK) CONTAINS 1048576 LOCKS.
THE CONTENTION NOTIFYING SYSTEM IS SC63
SYNCHRES:       NO
```

Setting up MIM multi-image integrity for DFSMSrmm

If you have installed MIM as opposed to GRS, you have to update your MIM definitions.

DFSMSrmm uses the RESERVE macro and ENQ macro to serialize use of its own resources. The SCOPE specified by DFSMSrmm ensures that serialization is across systems, within a system, or within a task, as required. All serialized resources use the SYSZRMM QNAME, and the RNAME is determined based on the resource required. If you do nothing to define the SYSZRMM QNAME, the serialization required by DFSMSrmm should be provided.

For a complete list of the resources serialized for DFSMSrmm, and the RNAMES, and SCOPE of each request refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

To serialize the DFSMSrmm CDS the RESERVE macro with a RNAME of MASTER.RESERVE is used, which results in:

- A hardware reserve
- A SYSTEMS enqueue

In a multi-system environment, the SYSTEMS enqueue is not required as well a hardware reserve, and there may also be times when a hardware reserve is not required because of other data on the same volume. You can optionally use MIMQNAME member entries to optimize the DFSMSrmm serialization for your environment.

If you want to use hardware reserves, we recommend the MIMQNAME member of the CNTL data set shown in Figure 3-21.

```
SYSZRMM    GDIF=YES,        /*  GDIF SHOULD PROCESS THIS QNAME
           SCOPE=SYSTEMS,   /*  Identify which ENQUEUES to manage
           RESERVES=KEEP,   /*  Use Hardware reserves if requested
           EXEMPT=YES,      /*  EXEMPT LIST is required for SCOPE=SYSTEMS
           ECMF=NO,         /*  ECMF not required
           TRACE=NONE       /*  Do not trace requests
```

Figure 3-21 MIMQNAME member using hardware reserves

The SCOPE=SYSTEMS ENQs issued by DFSMSrmm should be propagated to ensure correct serialization, except for the ENQ related to the hardware reserve.

The EXEMPT=YES identifies that one ENQ should not be propagated; you specify these in the GDIEXMPT member (see Figure 3-22).

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*  DFSMSrmm LOCAL EXCLUSION STATEMENTS. DO NOT PROPAGATE ENQS       *
*  FOR CDS SERIALIZATION.                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
LOCAL    QNAME=SYSZRMM,RNAME=MASTER.RESERVE
```

Figure 3-22 GDIEXMPT member

If you do not want to use hardware reserves, and will rely on propagation of the ENQs that serialize the DFSMSrmm CDS, we recommend the MIMQNAME member of the CNTL data set shown in Figure 3-23.

```
SYSZRMM   GDIF=YES,           /* GDIF SHOULD PROCESS THIS QNAME
          SCOPE=SYSTEMS,      /* Identify which ENQUEUES to manage
          RESERVES=CONVERT,   /* Use Hardware reserves if requested
          EXEMPT=YES,         /* EXEMPT LIST is required for SCOPE=SYSTEMS
          ECMF=NO,            /* ECMF not required
          TRACE=NONE          /* Do not trace requests
```

Figure 3-23 MIMQNAME member using propagation of ENQs

You can optionally specify other keyword values, such as ECMF=YES or TRACE=CONFLICT or TRACE=ALL depending on your local requirements and whether you are testing with DFSMSrmm or running in parallel.

3.1.4 Creating a starting procedure

System symbols are elements that allow systems to share PARMLIB definitions while retaining unique values in those definitions. You can code the statements in the IEASYMxx member of PARMLIB, so only one IEASYMxx member is needed to define static system symbols and system parameters for all systems in a multisystem environment. If you wish, you can code additional IEASYMxx members to organize the definitions more clearly.

If you have to define DFSMSrmm on multiple systems, you can check the symbols defined in your system before you create the procedure. To check all the defined symbols in an IEASYMxx member, you can use the MVS DISPLAY SYMBOLS command as shown in Figure 3-24.

Archived

```
DISPLAY SYMBOLS
```

Figure 3-24 Display defined system symbols

In Example 3-7 you can see all the available system-wide symbols on your system.

Example 3-7 Display SYMBOLS output

```
D SYMBOLS
IEA007I STATIC SYSTEM SYMBOL VALUES 059
        &SYSALVL.  = "2"
        &SYSCLONE. = "65"
        &SYSNAME.  = "SC65"
        &SYSPLEX.  = "SANDBOX"
        &SYSR1.    = "Z05RA1"
        &ALLCLST1. = "CANCEL"
        &CMDLIST1. = "65,00"
        &COMMDSN1. = "COMMON"
        &CPENABLE. = "(10,30)"
        &DCEPROC1. = "."
        &DFHSMCMD. = "00"
        &DFHSMHST. = "5"
        &DFHSMPRI. = "NO"
        &DFSPROC1. = "."
        &DLIB1.    = "Z05DZ1"
        &DLIB2.    = "Z05DZ2"
        &IEFSSNXX. = "R7"
        &IFAPRDXX. = "4A"
        &IGDSMSXX. = "02"
        &IXFPMOD1. = "TZ05J3"
        &J2PRM.    = "G"
        &LPALIST1. = "5A"
        &MLPALST1. = "0E"
        &OMVSPARM. = "5A"
        &PLPADSN1. = "PLPA"
        &PRISUBSY. = "JES3"
        &PROCLIB1. = "TEST1"
        &PROGXXA1. = "A0"
        &PROGXXE1. = "E5,E9"
        &PROGXXL1. = "D0"
        &PROGXXL2. = "5A,43"
        &PROGXXP1. = "P0"
        &PROGXXS1. = "S0"
        &RMFMON01. = "XA"
        &RSUVALUE. = "0"
        &SDSFPARM. = "3A"
        &SECSUBSY. = "JES2"
```

Archived

```
&SMFPARMS. = "00"
&SRES2.    = "Z05RA2"
&SYSR2.    = "Z05RA2"
&SYSR3.    = "Z05RA3"
```

Using the Problem Determination Aid facility

To analyze DFSMSrmm running problems you can use the Problem Determination Aid (PDA) facility to gather related information. The PDA facility is required for IBM service, because it traces module and resource flow. The PDA facility consists of in-storage trace, optional DASD log data sets, EDGRMMxx PARMLIB member options, and operator commands to control tracing.

If you need an external DASD record of trace data, perform this step once for each MVS image.

Two separate log data sets are used by the PDA facility. DFSMSrmm recognizes these log data sets by using EDGPDOX and EDGPDOY DD statements. The EDGPDOX and EDGPDOY data sets must be pre-allocated and cataloged to be used by DFSMSrmm. Both data sets must be allocated on the same volume. The DFSMSrmm RACF user ID requires ALTER access to the EDGPDOX and EDGPDOY data sets.

Before you can use the PDA facility, you need to:

- Determine how long you want to keep trace information.
- Optionally allocate storage on DASD for the PDA log data sets, EDGPDOX and EDGPDOY.
- Implement the PDA facility based on how long you want to keep trace data.

Refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405, for detailed information about the PDA facility.

Create a procedure in SYS1.PROCLIB to start the DFSMSrmm subsystem address space. Example 3-8 shows the sample JCL.

Example 3-8 Sample DFSMSrmm starting procedure

```
//DFRMM   PROC M=00,OPT=MAIN
//IEFPROC EXEC PGM=EDG&OPT.,PARM='&M',TIME=1440,REGION=0M
//PARMLIB DD   DDNAME=IEFRDER
//IEFRDER DD   DISP=SHR,DSN=SYS1.PARMLIB
//EDGPDOX DD   DISP=SHR,DSN=RMM.&SYSNAME..RMMPDOX
//EDGPDOY DD   DISP=SHR,DSN=RMM.&SYSNAME..RMMPDOY
```

In this example we use REGION=0M to get the maximum region size. If your default region size is small, we recommend you specify REGION=40M.

In this sample JCL we use:

DFRMM Specifies the procedure name. You can use any procedure name from 1 to 8 characters long.

M Allows you to specify a PARMLIB member suffix. M=00 tells DFSMSrmm to use EDGRMM00.

OPT Allows you to specify:
OPT=RESET to disable the subsystem interface
OPT=MAIN to execute the DFSMSrmm subsystem.

Note: Before DFSMSrmm disables the subsystem interface, it ensures that the user who made the request is authorized. In this case, the RACF user of the STC needs ALTER access the STGADMIN.EDG.RESET.SSI resource defined in RACF FACILITY class.

EDGPDOX An optional statement required only for an external trace output recording. If not specified, no logging of the PDA trace output is performed. The data set specified must be preallocated and cataloged. DFSMSrmm RACF user ID requires ALTER authority to this data set. This must be allocated on the same volume as EDGPDOY.

EDGPDOY An optional statement required only for external trace output recording. There are the same considerations as for EDGPDOX.

&SYSNAME This specifies a system symbol to use the same procedure on different systems.

Note: You must define a pair of trace output data sets for each MVS image. Do not share the trace data sets with multiple DFSMSrmm systems or with other system components.

For special considerations or restrictions when coding the starting procedure, refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

3.1.5 Define an alias for high-level qualifier RMM

We recommend that you to create an alias in the ICF user catalog for RMM as shown in Example 3-9.

Archived

Example 3-9 Define an alias in an ICF-user catalog

```
//DEFALIAS  JOB   ...
//STEP1     EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
   DEFINE ALIAS                    -
          (NAME(RMM)               -
          RELATE('SYSCAT1.SYST01'))
```

The DEFINE ALIAS command defines an alias, RMM, for the user catalog SYSCAT1.SYST01. All catalog entries for data sets defiled with a high-level qualifier of RMM will be defined in user catalog SYSCAT1.SYST01.

3.1.6 Protecting DFSMSrmm resources

To protect DFSMSrmm functions, you need to use an external security product, such as RACF. Invocation to this product should be through the security authorization facility (SAF). If RACF is not installed, you must provide equivalent function through the SAF interface.

The security implementation tasks are as follows:

- Assign a RACF user ID to DFSMSrmm
- Identify DFSMSrmm to RACF
- Define RACF groups for DFSMSrmm
- Define DFSMSrmm resources to RACF
- Check the STGADMIN.ADR.DUMP.CNCURRNT resource

Before you start DFSMSrmm, you need to create a RACF user ID for DFSMSrmm and define the DFSMSrmm started task to RACF. You can find more detailed information about how to implement RACF security for DFSMSrmm, and sample definitions in Appendix C, “Security topics” on page 647.

Assigning DFSMSrmm a RACF user ID

You can assign a RACF user ID that matches the name of the DFSMSrmm procedure you created in 3.1.4, “Creating a starting procedure” on page 85, but any installation-selected RACF user ID is acceptable. As data sets are created for use by the DFSMSrmm procedure, add the DFSMSrmm RACF user ID to the access list for the data sets.

Figure 3-25 shows the RACF command we used to define the DFSMSrmm user ID.

```
ADDUSER DFRMM DFLTGRP(SYS1) NAME('DFSMSrmm Userid')
```

Figure 3-25 Add DFSMSrmm user ID to RACF

Note: If you do not specify the DFLTGRP parameter on the ADDUSER command, the current connect group of the user issuing the ADDUSER command is used as the default.

Identifying DFSMSrmm procedures to RACF

To assign RACF identities to a started procedures, you can use the RACF STARTED class to add or modify security definitions for new and existing started procedures.

Issue the RACF commands in Figure 3-26 to assign RACF identities to the DFSMSrmm started procedure.

```
RDEFINE STARTED (DFRMM.*) UACC(NONE) STDATA(USER(DFRMM) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
```

Figure 3-26 Assign DFSMSrmm to RACF STARTED class

Note: Remember to refresh the in-storage profiles, using the SETROPTS REFRESH command, after you add profiles to the STARTED class.

If you plan to use the EDGLABEL, EDGXPROC, or the EDGBKUP procedures, you must define the procedures in the RACF STARTED class.

Before RACF 2.1 no STARTED class was available for use, the only way to associate a started procedure with a RACF user ID was by coding the RACF started procedures table, ICHRIN03.

3.1.7 Defining RACF groups for DFSMSrmm

To implement RACF security for DFSMSrmm resources, you should define a set of user groups to handle different kinds of resources at different access levels. For more information, refer to Appendix C, “Security topics” on page 647.

STGADMIN.EDG.RESET.SSI FACILITY class profile

At this stage there is one security profile in particular that you should be aware of and create. When you first install and start to use DFSMSrmm, it is likely that you will need to stop or even remove DFSMSrmm from your system. Although you can easily stop DFSMSrmm, if you have EDGSSSI in the subsystem name table DFRM entry, or have been using a running mode that supports tape recording, DFSMSrmm prevents all tape usage until it is restarted. To allow tapes to be used, you must restart DFSMSrmm, or remove it from the system.

The RACF FACILITY class profile STGADMIN.EDG.RESET.SSI controls the use of the RESET facility for removing DFSMSrmm from the system. Be sure to define this security profile and authorize the DFSMSrmm started task to reset the subsystem interface. Grant the DFSMSrmm started task procedure RACF user ID ALTER access authority to STGADMIN.EDG.RESET.SSI. Refer to the manual *DFSMSrmm Implementation and Customization Guide*, SC26-7405, for specific details on removing DFSMSrmm from the system.

Note: You can use the RESET facility without defining this profile when you do not have a security product installed.

3.1.8 Create the DFSMSrmm control data set (CDS)

The DFSMSrmm CDS is a VSAM KSDS that contains the complete inventory of the removable media library. DFSMSrmm records all changes made to the inventory, such as adding or deleting volumes in the CDS.

You can define the DFSMSrmm control data set as either an extended format (EF) or a non-extended format VSAM data set. Using a non-extended format data set for the DFSMSrmm control data set, limits the control data set size to a maximum of 4 GB. Using an EF data set enables you to use VSAM functions such as multivolume allocation, compression, or striping. EF also enables you to define a control data set that uses VSAM extended addressability (EA) to enable the control data set to grow above 4 GB.

To define an EF control data set, you must include the DATACLASS keyword on the AMS DEFINE command and reference the correct data class. Refer to *z/OS DFSMS: Using Data Sets,* SC26-7410 for more information on EF data sets, and refer to *DFSMSdfp Storage Administration Reference*, SC26-7402 for information on defining data classes with DSNTYPE=EXT and EXTENDED ADDRESSABILITY=Y.

DFSMSrmm requires CONTROL access to the control data set. The control data set cannot be a SYS1.xx data set if the control data set is to be shared.

Additionally, batch LSR cannot be used with the DFSMSrmm control data set. In our example, we will use the non-extended format.

To create the CDS on the system, perform the following steps:

1. Calculate the space for the DFSMSrmm CDS.
2. Allocate the DFSMSrmm CDS.
3. Protect the CDS.
4. Initialize the CDS using the EDGJUTIL utility.

Calculate the space for the DFSMSrmm CDS

There is sample JCL in SAMPLIB member EDGJMFAL to allocate the CDS. You can modify it to meet your installation standards.

Since the control data set is one of the most important parts of DFSMSrmm, to avoid space problems while running DFSMSrmm, you must calculate the space required before its allocation.

Table 3-1 shows the space requirement for each kind of record in the CDS.

Table 3-1 DFSMSrmm CDS space requirement

Control data set content	DASD space
Control record	1 MB (MB equals approximately 1 000 000 bytes)
Data sets	512 KB (KB equals approximately 1000 bytes) for every 1000 data sets
Shelf locations in the library that do not contain volumes	140 KB for every 1000 shelf locations
Shelf locations in storage locations	140 KB for every 1000 shelf locations
Owners	38 KB per 1000 volumes
Software products, average five volumes per product	420 KB for every 1000 software products
Volumes	1 MB for every 1000 volumes
Vital record specifications	212 KB for every 1000 vital record specifications

Allocate the DFSMSrmm CDS

The JCL to create the DFSMSrmm CDS in Example 3-10 has been modified to allocate a bigger INDEX than that calculated by VSAM.

Example 3-10 Sample JCL to create the DFSMSrmm CDS

```
//LCLCDSAL JOB ,14O.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************** *
//*          STEP 1  Delete old DFSMSrmm control data set          *
//*               2  Allocate a new DFSMSrmm control data set      *
//* ************************************************************** *
//STEPO1   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.PROD.CDS CL
  SET MAXCC = 0
/*
//STEPO2   EXEC PGM=IDCAMS                  ALLOCATION MASTER FILE
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,UNIT=3390,VOL=SER=DFRMM4
//SYSIN    DD   *
  DEFINE CLUSTER (NAME(RMM.PROD.CDS)                -
                  FILE(MASTER)                      -
                  FREESPACE(15 0)                   -
                  KEY(56 0)                         -
                  REUSE                             -
                  RECSZ(512 9216)                   -
                  SHR(3 3)                          -
                  STORCLAS(storage_class)           -
                  VOLUMES(DFRMM4))                  -
          DATA    (NAME(RMM.PROD.CDS.DATA)          -
                  CYLINDERS(300 300)                -
                  CISZ(18432)                       -
                  FILE(MASTER))                     -
          INDEX   (NAME(RMM.PROD.CDS.INDEX)         -
                  CYLINDERS(30 30)                  -
                  FILE(MASTER))
/*
```

Note: For initial testing with DFSMSrmm, a small CDS will do. When you convert your existing data to DFSMSrmm, you will need a CDS sized to accommodate your existing and planned requirements. See 7.6, “DFSMSrmm CDS creation” on page 302 for more information about how to create a CDS for conversion.

Protect the CDS

To protect the DFSMSrmm CDS, define a discrete profile, and then permit the appropriate RACF groups to the RACF data set profile. Example 3-11 contains sample JCL. For detailed information on implementing RACF security for DFSMSrmm, refer to Appendix C, “Security topics” on page 647.

Archived

Example 3-11 Sample JCL to protect the DFSMSrmm CDS

```
//RACFDSN  JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************** *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /*   DEFINE CONTROL DATA SET DATA SET PROFILE                     */
 /* ************************************************************** */
   ADDSD 'RMM.PROD.CDS' UACC(NONE) GENERIC
      PERMIT 'RMM.PROD.CDS' GENERIC-
             ID(EDGADMIN) ACC(READ)
      PERMIT 'RMM.PROD.CDS' GENERIC-
             ID(EDGLIB) ACC(READ)
      PERMIT 'RMM.PROD.CDS' GENERIC-
             ID(EDGSYSPG) ACC(ALTER)
```

Initialize the CDS using the EDGJUTIL utility

Before using the DFSMSrmm CDS, a control record must be written in the CDS using the EDGUTIL utility. The control record contains information about the number of shelf locations in the library and storage locations.

Sample JCL is provided in SYS1.SAMPLIB member EDGJUTIL, which you can modify to meet your installation standards. Example 3-12 shows sample JCL that can be used to initialize the CDS.

Example 3-12 Sample JCL to initialize the DFSMSrmm CDS

```
//EDGUTIL  JOB ,14O.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//*********************************************************************
//*         STEP 1  Initializing the Control Data Set                 *
//*********************************************************************
//EDGUTIL  EXEC PGM=EDGUTIL,PARM='CREATE'
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   SYSOUT=*
//MASTER   DD   DISP=OLD,DSN=RMM.PROD.CDS
//SYSIN    DD   *
  CONTROL EXTENDEDBIN(YES) CATSYNCH(NO)
/*
```

Archived

Where:

CATSYNCH(NO) Specify CATSYNCH(NO) to force synchronization of the DFSMSrmm control data set and user catalogs the next time inventory management is run. See 1.1.9, “Catalog synchronization” on page 41 for more information.

EXTENDEDBIN(YES) Enables DFSMSrmm extended bin support, which allows the reuse of bins at the start of a move.
Extended bin support must be enabled if you want to use the DFSMSrmm PARMLIB OPTION command REUSEBIN(STARTMOVE) operand as shown in Figure 3-4 on page 76 to reuse bins when a volume moves from a bin.

3.1.9 Create the DFSMSrmm CDS as extended format

You can define the DFSMSrmm CDS as an extended format (EF) VSAM data set. This function is enabled with APAR OW47639. The benefits available for extended format data sets include:

- Data striping
- Data compression
- VSAM extended addressability
- Partial space release
- System-managed buffering
- CDS above 4 GB

Extended format enables you to define a CDS that uses VSAM Extended Addressability (EA) to enable the CDS to grow above 4 GB.

Extended format data sets must be system-managed. The mechanism to request extended format is through the SMS data class attribute `Data Set Name Type`, specifying a value of `EXT`. For an extended format data set to be allocated using extended addressability, the attribute `Extended Addressability ... Y` (see Figure 3-27).

Archive

```
                          DATA CLASS DEFINE                 Page 2 of 4
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : EFKSDS

To DEFINE Data Class, Specify:

  Data Set Name Type  . . . . . . EXT  (EXT, HFS, LIB, PDS or blank)
    If Ext  . . . . . . . . . . . R    (P=Preferred, R=Required or blank)
    Extended Addressability . . . Y    (Y or N)
    Record Access Bias  . . . . .      (S=System, U=User or blank)
  Space Constraint Relief . . . . N    (Y or N)
    Reduce Space Up To (%)  . . .      (0 to 99 or blank)
    Dynamic Volume Count  . . . .      (1 to 59 or blank)
  Compaction  . . . . . . . . . .      (Y, N, T, G or blank)
  Spanned / Nonspanned  . . . . .      (S=Spanned, N=Nonspanned or blank)
```

Figure 3-27 Data class specifying extended format

To define an extended format or extended addressability DFSMSrmm CDS, you must include the DATACLAS keyword as shown in Example 3-13, and reference the correct data class. For more information about how to reference a data class, refer to *DFSMS Access Method Services for Catalogs*, SC26-7394.

Example 3-13 Sample JCL to create an extended format DFSMSrmm CDS

```
//LCLCDSAL JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************** *
//*         STEP 1  Delete old DFSMSrmm control data set           *
//*              2  Allocate a new DFSMSrmm control data set       *
//* ************************************************************** *
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.PROD.CDS CL
  SET MAXCC = 0
/*
//STEP02   EXEC PGM=IDCAMS                  ALLOCATION MASTER FILE
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,UNIT=3390,VOL=SER=DFRMM4
//SYSIN    DD   *
  DEFINE CLUSTER (NAME(RMM.PROD.CDS)                  -
                  FILE(MASTER)                        -
                  FREESPACE(15 0)                     -
                  KEY(56 0)                           -
                  REUSE                               -
```

```
                RECSZ(512 9216)                 -
                SHR(3 3)                        -
                STORCLAS(storage_class)         -
                DATACLASS(EFKSDS)               -
                VOLUMES(DFRMM4))                -
        DATA    (NAME(RMM.PROD.CDS.DATA)        -
                CYLINDERS(300 300)              -
                CISZ(18432)                     -
                FILE(MASTER))                   -
        INDEX   (NAME(RMM.PROD.CDS.INDEX)       -
                CYLINDERS(30 30)                -
                FILE(MASTER))
/*
```

3.1.10 Create the DFSMSrmm journal

The DFSMSrmm journal contains a record of all changes made to the CDS since the last backup. We recommend that you create the journal and use it to forward recover changes made since the last backup. Each time you successfully back up the CDS by using the EDGHSKP utility, the journal data set is cleared.

We recommend that you allocate the journal data set on a different volume than the CDS, back up the journal data set, and maintain multiple generations of it.

To create the DFSMSrmm journal on the system, perform the following steps:

1. Calculate the space of the DFSMSrmm journal.
2. Allocate the DFSMSrmm journal.
3. Protect the DFSMSrmm journal.

Calculate the space of the DFSMSrmm journal

Table 3-2 shows the space requirement for each kind of record in the journal data set.

Table 3-2 Space requirement for journal data set record

Journal Content	DASD Space
Changes by users	1.5 KB for each change made
Data sets	1.5 KB for each data set retained by a vital record specification
Data sets no longer retained by a vital record specification	1.5 KB for each data set no longer retained by a vital record specification
Expiring volumes 1.5 KB for each expiring volume	1.5 KB for each expiring volume

Archived

Journal Content	DASD Space
Non-scratch mounts	6.7 KB for each mount
Scratch mounts	8.3 KB for each mount
Volumes	1.5 KB for each volume retained by a vital record specification
Volume checked in/out	2.6 KB for each volume in or out of the library
Volumes returned to scratch	3.0 KB for each volume returned to scratch
Volumes to and from storage locations	3.5 KB for each volume moved to or from a storage location
Volumes no longer retained by a vital record specification	1.5 KB for each volume that has not reached its expiration date and is no longer retained by a vital record specification
Volumes that are exported or imported	1.5 KB for each logical volume exported or imported
Vital record specifications	1.3 KB for each vital record specification created

Divide the total KB of space by 4, because allocation will be by average record size using a 4 K value. Use this number to in the space allocation (SPACE=) shown in Example 3-14.

Allocate the DFSMSrmm journal

Sample JCL to create the DFSMSrmm journal is provided in SYS1.SAMPLIB member EDGJNLAL. You can modify the sample JCL to meet your installation standards. Example 3-14 shows sample JCL to create the DFSMSrmm journal.

Example 3-14 Sample JCL to create the DFSMSrmm journal

```
//LCLJRNLA JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************* *
//*          STEP 1  DELETE OLD DFSMSRMM JOURNAL                  *
//*                  2  ALLOCATE A NEW DFSMSRMM JOURNAL           *
//* ************************************************************* *
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DEL RMM.PROD.JRNL
 SET MAXCC=0
/*
//STEP02   EXEC PGM=IEFBR14
```

```
//SYSPRINT DD   SYSOUT=*
//JOURNAL  DD   DISP=(,CATLG,DELETE),
//         DSN=RMM.PROD.JRNL,
//         UNIT=3390,VOL=SER=DFRMM3,
//         AVGREC=U,SPACE=(4096,(6957),,CONTIG)
```

For initial testing with DFSMSrmm, either no journal or a small journal will do. Later, when you use converted data, you will need to use a journal that is large enough for your daily needs.

Protect the DFSMSrmm journal

To protect the DFSMSrmm journal, define a discrete profile, and then permit the appropriate RACF groups to the RACF data set profile. Example 3-15 contains sample JCL. For detailed information on implementing RACF security for DFSMSrmm, refer to Appendix C, “Security topics” on page 647.

Example 3-15 Sample JCL to protect the DFSMSrmm CDS

```
//RACFDSN  JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ******************************************************************* *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ******************************************************************* *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ****************************************************************** */
 /*   DEFINE JOURNAL DATA SET DATA SET PROFILE                         */
 /* ****************************************************************** */
   ADDSD 'RMM.PROD.JRNL' UACC(NONE) GENERIC
      PERMIT 'RMM.PROD.JRNL' GENERIC-
             ID(EDGADMIN) ACC(READ)
      PERMIT 'RMM.PROD.JRNL' GENERIC-
             ID(EDGLIB) ACC(READ)
      PERMIT 'RMM.PROD.JRNL' GENERIC-
             ID(EDGSYSPG) ACC(ALTER)
```

Archived

3.1.11 Restart MVS with DFSMSrmm implemented

Recommendation: You can schedule the IPL to implement the DFRM subsystem any time ahead of migration and parallel running, but you need to issue the command **`S DFRMM,OPT=RESET`** as part of that IPL before tapes are used.
You can use this command only if you have defined the started task DFRMM as described in Example 3-8 on page 87, and you have defined the resource STGADMIN.EDG.RESET.SSI in RACF class FACILITY as described in 3.1.7, “Defining RACF groups for DFSMSrmm” on page 90. The RACF user ID of the started task needs ALTER access to this profile.

You are ready to start the system with DFSMSrmm implemented. You may be able to dynamically implement changes to the MVS system PARMLIB members or modified installation exits depending on your MVS level. You must:

- IPL with CLPA to include new levels of the DFSMSrmm code that have LPALIB as the target library
- IPL when you make changes to IFAPRDxx to change the licensing options.
- IPL to implement the DFRM subsystem if you plan to run DFSMSrmm in parallel with any other tape management system
- You can implement DFSMSrmm without the need of an IPL if you use the TSO command (see Figure 3-9 on page 65) and MVS commands (see Figure 3-2 on page 63) as described previously.

Important: If you have installed a z/OS release prior to z/OS V1.3 and TLMS is running, you cannot update the subsystem name table dynamically to implement DFSMSrmm. In this case, you have to re-IPL your MVS system with CLPA.
If you have installed DFSMSrmm Release 2.10, you can update the subsystem name table dynamically to implement DFSMSrmm, but there will be error messages like `EDG0003E` or `EDG0004E`; these can be ignored. When `EDG0103D` prompts, reply `RETRY`.

With z/OS V1.3 and following releases, there is no longer a requirement to have DFSMSrmm started before other tape management systems. DFSMSrmm does not update LPA or use dynamic hooks into DFSMSdfp. Therefore, the only need for an IPL with CLPA beginning with z/OS V1.3 is if any DFSMSrmm maintenance goes into LPALIB.

3.1.12 Starting and restarting DFSMSrmm

Usually, the DFSMSrmm subsystem starts automatically through standard initial program load (IPL). Under some conditions, you might need to start or restart the subsystem.

To start DFSMSrmm using the default procedure name and PARMLIB member, enter at the MVS command, as shown in Example 3-16.

Example 3-16 MVS command to start DFSMSrmm

```
S DFRMM
```

If you want, start DFSMSrmm with different parameters than the default; you can use the MVS command as shown in Example 3-17.

Example 3-17 VS command to start DFSMSrmm with parameters

```
S DFRMM,M=xx,DSN=parmlib_dataset
```

Where:

M=xx	Specify the suffix of the EDGRMMxx member in PARMLIB.
DSN=	Specify a different PARMLIB data set name|

After this command, you must see the message in Figure 3-28.

```
EDG0105I DFSMSrmm SUBSYSTEM INITIALIZATION COMPLETE
```

Figure 3-28 EDG0105I message after initialization of the SSI

If the subsystem interface was not activated before the command, you can get the message shown in Figure 3-29.

```
EDG0103D DFSMSrmm SUBSYSTEM INTERFACE IS INACTIVE - ENTER "IGNORE",
         "CANCEL" OR "RETRY"
```

Figure 3-29 EDG0103D message to activate the SSI

Enter `RETRY` to continue with the initialization; after that you must receive the `EDG0105I` message.

You can restart DFSMSrmm using the MVS **`MODIFY`** command. For example:

- To restart DFSMSrmm and implement new PARMLIB options, specify the command shown in Example 3-18.

Example 3-18 Modify DFSMSrmm using new PARMLIB options

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

- To turn Problem Determination Aid (PDA) tracing on or off, specify one of the commands shown in Example 3-19.

Example 3-19 Turn PDA on of off

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

- To manually quiesce DFSMSrmm, specify the command shown in Example 3-20.

Example 3-20 Quiesce all DFSMSrmm activities

```
F DFRMM,QUIESCE
```

- To refresh DFSMSrmm installation exits and enable any exits that are currently disabled, specify the command shown in Example 3-21.

Example 3-21 Activate or refresh DFSMSrmm installation exits

```
F DFRMM,REFRESH EXITS
```

For more information about these commands, see *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

3.1.13 Stopping DFSMSrmm

The DFSMSrmm subsystem must always be activated in your system. In some recovery situations, you might need to remove DFSMSrmm from your system. In that case, you must follow the following procedure:

- All jobs that are processing and using tapes should be completed. Before stopping DFSMSrmm, you can stop batch initiators to avoid failing a job that opens tape data set.
  Before stopping DFSMSrmm, you can obtain a list of the number of requests waiting to be processed and the number of active requests. You use the command shown in Example 3-22.

Example 3-22 Show all outstanding DFSMSrmm requests

```
F DFRMM,QUERY ACTIVE
```

As a result of this command, you will see the messages shown in Figure 3-30.

```
EDG1113I F=function requestor_type=requestor_name time TKN=token_value
EDG1114I COMMAND COMPLETE, TOTAL TASKS total_count, ACTIVE active_count
EDG1118I queued_count QUEUED REQUESTS, INCLUDING nowait_count NOWAIT
         catalog_count CATALOG
```

Figure 3-30 Results of F DFRMM,QUERY ACTIVE command

where:

F	Identifies the requested DFSMSrmm function. The values are internal to DFSMSrmm
Requestor_type	Identifies the requestor. It can be JOB, STC, or TSU.
Time	Lists the time that the request was started
TKN	Identifies the request
Total_count	Is the number of tasks available to process DFSMSrmm requests
Active_count	Is the number of tasks currently processing
Queued_count	Is the number of requests waiting to be processed by DFSMSrmm
Nowait_count	Is the number of requests waiting to be processed by DFSMSrmm for which the request does not wait for the results. Includes the count of catalog requests.
Catalog_count	Is the number of requests waiting to be processed by DFSMSrmm to reflect catalog update activity in the DFSMSrmm control data set

In Figure 3-32 you can see an example output from this command. In this sample, the only request to DFSMSrmm is the housekeeping processing executed by the YCJRES1B job.

```
EDG1101I DFSMSrmm MODIFY COMMAND ACCEPTED
EDG1119I CURRENT DFSMSrmm STATUS IS ACTIVE
EDG1113I F=130      JOB=YCJRES1B 22:19:54 TKN=04000033
EDG1114I COMMAND COMPLETE, TOTAL TASKS 10, ACTIVE  1
EDG1118I      0 QUEUED REQUESTS, INCLUDING      0 NOWAIT      0 CATALOG
```

Figure 3-31 DFSMSrmm MODIFY command output

- Optionally, you may need to remove the DFSMSrmm subsystem. Usually, you do not need to remove it. If you want to allow tape use without DFSMSrmm recording, then you can use the command shown in Figure 3-23.

Example 3-23 Disable the DFSMSrmm SSI

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

Note: To use this command, the user ID of the started task needs ALTER access to the resource STGADMIN.EDG.RESET.SSI defined in the RACF FACILITY class.

Then, wait for the response shown in Figure 3-32.

```
IEF403I DFRMM - STARTED - TIME=18.48.27 - ASID=0063.
EDG0181I DFSMSRMM SUBSYSTEM INTERFACE SUCCESSFULLY INACTIVATED
```

Figure 3-32 S DFRMM,OPT_RESET command output

The use of OPT=RESET must be RACF protected. Refer to Appendix C.1.3, “Define DFSMSrmm resources to RACF” on page 650 for additional information about the RACF profiles.

Important: The OPT=RESET operand should only be used during parallel running or testing, when DFSMSrmm must be completely removed from the system due to an error, or because you need to reconvert.

- Finally, shut down the DFSMSrmm subsystem, using the command shown in Figure 3-24.

Example 3-24 Stopping the DFSMSrmm subsystem

```
P DFRMM
```

When you get the following messages shown in Figure 3-33, DFSMSrmm is completely removed from your system.

```
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 DFRMM    ENDED
```

Figure 3-33 P DFRMM command output

3.1.14 Setting up DFSMSrmm utilities

There are several DFSMSrmm utilities that you should set up now:

- EDGHSKP is used for performing the following inventory management activities:

  - Vital record processing, to determine which volumes to retain and which volume moves are required, based on VRSs
  - Expiration processing to identify volumes ready to be released and returned to scratch
  - Storage location management processing to assign shelf locations to volumes being moved to storage locations
  - Backup and recovery of the CDS and journal:
    - Use the DFSMSrmm backup utilities rather than other backup utilities, such as the IDCAMS EXPORT command, because DFSMSrmm provides the necessary serialization and forward recovery functions. DFSMSrmm backup utilities check whether the CDS is in use, tell the DFSMSrmm subsystem that backup or recovery is in process, and provide a way to forward recover.
  - Creation of an extract data set. You can produce movement and inventory reports by producing an extract data set from the CDS, and creating a report from it with the report utility.

- EDGAUD and EDGRPTD can help you in getting information about your removable media library and storage locations. You can also get security related information about volumes and data sets defined to DFSMSrmm, and audit trail information about volumes, shelf assignments, and user activity.
- EDGUTIL is the utility used to create, update, and verify the CDS.
- EDGINERS is the DFSMSrmm utility that helps you erase and initialize tape volumes either automatically or manually. You can use EDGINERS to replace the DFSMSdfp utility IEHINITT.

You should create a sample schedule of how frequently to run DFSMSrmm utilities. For more detailed information about the DFSMSrmm utilities, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

3.1.15 Running the installation verification program

The quickest and simplest way of getting DFSMSrmm started and verifying that the key pieces are installed successfully is to run the installation verification procedure (IVP). The IVP is documented in detail in *DFSMSrmm Implementation and Customization Guide*, SC26-7405. The information includes considerations for running in an environment where another tape management system is in use.

3.2 Tailoring the DFSMSrmm ISPF dialog

In this section, we describe how to make the DFSMSrmm ISPF panels available to the user, and how to modify the panels layout only for end users and librarians.

To make DFSMSrmm ISPF dialog available in the ISPF Master Application Menu, use one of the following techniques:

- Concatenate the DFSMSrmm target ISPF libraries with your existing ISPF libraries and use one of the following methods:
  - Add DFSMSrmm to the ISPF dialog as in 3.2.1, “Adding DFSMSrmm to ISPF panels” on page 106.
  - Use the default method supplied by DFSMS. Select option **R** from the Interactive Storage Management Facility (ISMF) primary option menu.
  - Use the RMMISPF EXEC to invoke the dialog.
- Use the ISPF LIBDEF facility to make the DFSMSrmm target ISPF libraries available to your users, and use the EDGRMLIB EXEC to enter the dialog.

  When using the LIBDEF facility, if you are using different target library names as listed in the *OS390 Program Directory* and *ServerPac: Installing Your Order*, you must modify the EDGRMLIB EXEC or produce your own similar REXX exec or CLIST.

3.2.1 Adding DFSMSrmm to ISPF panels

You can add DFSMSrmm selection in ISPF Main Application Menu, so that users can choose it easily. To add the selection:

- Add the DFSMSrmm selection to the body of the chosen ISPF selection panel. For example, add the text shown in Figure 3-34.

```
      RM RMM DFSMSrmm dialog
```

Figure 3-34 RMM ISPF selection

- Add one of the following statements to the ZSEL processing list in the)PROC section of the chosen ISPF panel:
  - If you are not using LIBDEF using the selection shown in Figure 3-35.

```
      R,'CMD(%RMMISPF) NEWAPPL(EDG)'
```

Figure 3-35 ISPF selection with NEWAPPL

– If you are using LIBDEF using the selection shown in Figure 3-36.

```
    R,'CMD(%EDGRMLIB)'
```

Figure 3-36 ISPF selection without NEWAPPL

Figure 3-37 shows an example definition using LIBDEF.

```
...
$AO @AOC     @- Automated Operations Control/MVS Dialogs      +
$RM @RMM     @- DFSMSrmm dialog                               +
$CN @CONS    @- Console Display and Search Facility           +
...
)PROC
&ZSEL =TRANS(
             ..
             AO,'CMD(%$AOC140)'
             RM,'CMD(%EDGRMLIB)'
             ..
....
```

Figure 3-37 Add DFSMSrmm to a ISPF selection panel

3.2.2 Modifying the DFSMSrmm panel

When you enter the DFSMSrmm ISPF dialog, usually the first panel you see is the Primary Option Menu panel. You can change the panel navigation for the different type of users to go directly to a lower level panel, so that when they enter DFSMSrmm, tape librarians always see the Librarian Menu and end users see the DFSMSrmm User Menu:

- You can modify the selection as follows, so that only the USER option of the DFSMSrmm dialog is available to the majority of end users. Choose one of the following ways to change:
  – When using LIBDEF, add to the)PROC section the selection shown in Figure 3-38.

```
    R,'CMD(%RMMISPF USER) NEWAPPL(EDG)'
```

Figure 3-38 RMM ISPF selection with NEWAPPL and USER selection

  – When using LIBDEF, modify the supplied EDGRMLIB EXEC to include the USER parameter on the RMMISPF EXEC call. For example, enter the following REXX statement as shown in Figure 3-39 to replace the one supplied in EDGRMLIB.

Archived

```
address "ISPEXEC""SELECT CMD(%RMMISPF USER) NEWAPPL(EDG)PASSLIB"
```

Figure 3-39 RMM ISPF USER selection using RMMISPF REXX exec

If you are not using LIBDEF, Table 3-3 shows the names of the default libraries that you concatenate to the DD statements in the TSO logon procedure or a user-supplied start-up CLIST. If you are using LIBDEF, the EDGRMLIB EXEC allocates these default libraries to the user. You need to modify the EDGRMLIB exec to contain the new library names, after you changed the names of the target libraries on your system.

Table 3-3 DFSMSrmm libraries and their default DD names

DFSMSrmm Data Set Name	DD Statement	Content
SYS1.SEDGEXE1	SYSPROC (or SYSEXEC)	REXX execs
SYS1.SEDGMENU	ISPMLIB	English messages
SYS1.SEDGPENU	ISPPLIB	English panel
SYS1.DGTSLIB	ISPSLIB	Skeletons
SYS1.DGTTLIB	ISPTLIB	Tables
SYS1.SEDGHLP1	SYSHELP	TSO HELP

- You can modify the selection as follows so that the LIBRARIAN option of the DFSMSrmm dialog is available to the librarians. Choose one of the following ways to change:
  - When using LIBDEF, add to the)PROC section using the selection shown in Figure 3-40.

```
R,'CMD(%RMMISPF LIBRARIAN)NEWAPPL(EDG)'
```

Figure 3-40 RMM ISPF selection with NEWAPPL and LIBRARIAN selection

  - When using LIBDEF, modify the supplied EDGRMLIB EXEC to include the USER parameter on the RMMISPF EXEC call. For example, enter the following REXX statement as shown Figure 3-41 to replace the one supplied in EDGRMLIB.

```
address "ISPEXEC""SELECT CMD(%RMMISPF LIBRARIAN)NEWAPPL(EDG)PASSLIB
```

Figure 3-41 RMM ISPF LIBRARIAN selection using RMMISPF REXX exec

Archived

3.2.3 Other changes

There are other changes you may want to implement in the DFSMSrmm ISPF dialog. These changes are:

- Add local dialog
  DFSMSrmm has a dummy panel named EDGP@LCL, which provides easy access to local dialog extensions. You can use these extensions without having to modify DFSMSrmm.
- Modify add product volume defaults.
  You can modify the dialog EXEC EDGRPADV to change the values to ones that suit your installation.

  Recommendation: Install your changes using an SMP/E USERMOD after editing a copy of the EDGRPADV EXEC.

- Modify DFSMSrmm messages.
  DFSMSrmm provides messages for report titles and user notification. You can customize them by:
  - Updating the text of a message in the DFSMSrmm message table EDGMTAB
  - Applying changes to EDGMTAB by creating an SMP/E-installable USERMOD

For more information, please refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

3.3 Making TSO HELP information available to users

DFSMSrmm provides online help for the RMM TSO command. The help information is, by default, installed into the SYS1.SEDGHLP1 data set, although this can be changed by you during the SMP/E installation process. To use the help information, copy the contents of SYS1.SEDGHLP1 to SYS1.HELP or concatenate the library to the other help libraries on the SYSHELP DD statement. You can update the TSO logon procedure or use the TSO ALLOCATE command. Figure 3-25 shows how to update the TSO logon procedure.

Example 3-25 Update the TSO logon procedure

```
//IKJACCNT PROC
//IKJACCT EXEC PGM=IKJEFT01, PARM='ISPPDF',
//    DYNAMNBR=150,TIME=1440
//SYSPROC  DD  DSN=SYS1.OS390.CLIST,DISP=SHR
//SYSHELP  DD  DSN=SYS1.HELP,DISP=SHR
```

```
//         DD  DSN=SYS1.SEDGHLP1,DISP=SHR
//ADMPC    DD  DSN=GDDM.SADMPCF,DISP=SHR
//ADMCGM   DD  DSN=GDDM.SADMSAM,DISP=SHR
//ADMGGMAP DD  DSN=GDDM.SADMMAP,DISP=SHR
//ADMPROJ  DD  DSN=GDDM.SADMMAP,DISP=SHR
//ADMIMG   DD  DSN=GDDM.SADMMAP,DISP=SHR
//ADMGIMP  DD  DSN=GDDM.SADMMAP,DISP=SHR
//ADMGDF   DD  DSN=GDDM.SADMMAP,DISP=SHR
//ADMSYMBL DD  DSN=GDDM.SADMSYM,DISP=SHR
//SYSPRINT DD  TERM=TS,SYSOUT=*
//SDFSDUMP DD  SYSOUT=*
//SYSTERM  DD  TERM=TS,SYSOUT=*
//SYSIN    DD  TERM=TS
//*
```

3.4 Verifying your DFSMSrmm implementation

To test DFSMSrmm, you can use a small, test CDS or a converted CDS. We recommend that you use simple DFSMSrmm function tests for your first testing. Later, after you have analyzed the data and customized the exits and parameters, you can test those functions that will be used in production.

Verifying DFSMSrmm functions by submitting batch jobs is quite simple. In this section we show you sample batch jobs we used to verify DFSMSrmm functions in our system. Run the sample jobs in the sequence shown to test basic DFSMSrmm functions in your system.

3.4.1 Display PARMLIB options and control information

Use the RMM LISTCONTROL subcommand to display information from the DFSMSrmm CDS control record and options defined in the EDGRMMxx PARMLIB member.

Example:

Display your installation's options and rules, restricting the information displayed to the control record information and system options only (see Example 3-26). Specify the ALL operand to display all options and rules defined to your installation.

Example 3-26 Sample JCL to display PARMLIB options and control information

```
//LISTCNTL JOB ,14O.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************* *
//*          STEP 1  DISPLAY INFORMATION                          *
```

```
//* ******************************************************************* *
//STEP01   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM LISTCONTROL ALL
/*
```

The output from the RMM LISTCONTROL command is shown in Example 3-27.

Example 3-27 Sample RMM LISTCONTROL subcommand output

```
Control record:
Type = MASTER    Create date = 1997/114    Create time = 14:28:32
Journal Utilization =  20% (75% threshold)      STATUS: = ENABLED
CDS:    Utilization =  75%
Exit status:                                Options:
  EDGUX100 = ENABLED                          Stacked Volumes        = NONE
  EDGUX200 = NONE                             Extended Bin           = DISABLED
Last backup:                                Last expiration processing:
  Date = 2003/252    Time = 16:58:03          Date = 2003/252    Time = 16:55:29
Last journal backup:                        Last store update:
  Date = 2003/252    Time = 16:58:03          Date = 2003/184    Time = 17:47:19
Last report extract:                        Last VRS processing:
  Date = 2003/196    Time = 15:02:00          Date = 2003/252    Time = 16:55:29
Last scratch procedure:                     Last Catalog synchronize:
  Date = 2000/006    Time = 05:10:23          Date =  2002/193   Time = 03:17:05
Rack numbers       = 2842                   Empty racks        = 1800
LOCAL store bins   = 244                    Empty LOCAL bins   = 244
DISTANT store bins = 200                    Empty DISTANT bins = 120
REMOTE store bins  = 999                    Empty REMOTE bins  = 999
Control functions in progress:
Backup         = N  Restore        = N
Verify         = N  Expiration     = N
Report Extract = N  Disaster Store = N
VRS            = N  Synchronize    = N

System options:
PARMLIB Suffix  = 00
Operating mode  = W    Retention period: Default = 5       Maximum = NOLIMIT
                                         Catalog = 12    hours
Control data set name      = RMM.PROD.CDS
Journal file data set name = RMM.PROD.JRNL
Journal threshold          = 75%
Catalog SYSID   = *
Scratch procedure name     = EDGXPROC
Backup procedure name      = EDGBKUP
IPL date check  = N        Date format    = J       RACF support   = A
SMF audit       = 248      SMF security   = 249     CDS id         = PROD
MAXHOLD value   = 100      Lines per page = 54      System ID      = EGZB
BLP             = RMM      TVEXT purge    = RELEASE Notify         = Y
Uncatalog       = Y        VRS job name   = 2       Message case   = M
MASTER overwrite= LAST     Accounting     = J       VRS selection  = NEW
```

```
VRS change      = INFO     VRSMIN action  = FAIL    VRSMIN count   = 10
Disp DD name    = LOANDD   Disp msg ID    = EDG4054I
Retain by       = VOLUME   Move by        = VOLUME  CMDAUTH Owner  = NO
PREACS          = NO       SMSACS         = NO      CMDAUTH Dsn    = YES
Reuse bin       = CONFIRMMOVE
PDA: OFF
 Block count    = 255        Block size     = 27       Log            = OFF
SMSTAPE:
 Update scratch = YES      Update command = YES     Update exits   = YES
 Purge          = ASIS

Security classes:
Number  Name      SMF  MSG  Erase  Description
------  --------  ---  ---  -----  --------------------------------
10      UCL       Y    N    N      UNCLASSIFIED
75      IUO       Y    N    N      IBM INTERNAL USE ONLY

Volume Pools:
Pool   System   RA Ty Expdt Pool     Media    Description
                CF pe check name     name
------ -------- -- -- ----- -------- -------- ------------------------------
F*              Y  R  Y             3480     Forign Cartridges
N*              N  R  0             3480     Test Pool for Norbert
PP*             Y  R  Y             3480     SOFTWARE Cartridges
SC*             Y  S  N             3590     SCRATCH 3480/3490 CASSETTES
*               Y  S  N             3480     ALL OTHER TAPES

Mount/Fetch messages:
Message ID    ID  Volume Rack
------------  --- ------ ----
IAT5110       1   44     999
IAT5210       1   50     999
IAT5410       1   20     999
IEC501A       1   15     999
IEC502E D     1   15     999
IEC502E K     1   15     999
IEC502E R     1   15     999
IEC502E RD    1   16     999
IEC502E RK    1   16     999
IEF233A       1   16     999
IEF233D       1   15     999
IEF234E D     1   15     999
IEF234E K     1   16     999
IEF234E R     1   15     999
IEF455D       1   15     999

Reject prefixes:
Rack Prefix  Access type
-----------  -----------
*            NONE

Location definitions:
Location Def Mgtype  Ltype  Priority Medianames
```

Archived

```
-------- --- ------  -----  -------- ----------
         N          AUTO     4800
         N          MANUAL   4900
DISTANT  N          STORE     200
IBMATL   Y          AUTO     4800
LOCAL    N          STORE     300
NEWYORK  Y          STORE     100   *
SAVE     Y  BINS    STORE    2000   *
SHELF    N                   5000
SANJOSE  Y  BINS    STORE    2000   *

Action   Status
-------- ---------
NOTIFY   Pending
RETURN   Pending
SCRATCH  Pending
REPLACE  Unknown
INIT     Unknown
ERASE    Unknown

Source   Target   Move Type Status
-------- -------- --------- ---------
DISTANT  ATL3494E NOTRTS    Complete
READY
```

3.4.2 Add owner information

Use the ADDOWNER subcommand to define an owner to DFSMSrmm. An owner can be an individual or group defined by a RACF group name, or any value you choose. The owner value contains one to eight alphanumeric characters.

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 requests the job as an DFSMSrmm owner ID. In all other cases the owner must be defined manually, and must be defined before you can use a RMM TSO subcommand that includes owner information.

Use the LISTOWNER subcommand to display information about a single owner defined to DFSMSrmm.

Example:

Add details of a new owner and in STEP02 request information recorded by DFSMSrmm about the owner whose owner ID is SCHLUM (see Example 3-28).

Archived

Example 3-28 Sample JCL to add owner information

```
//ADDOWNER JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************** *
//*          STEP 1  ADDING NEW OWNER                              *
//*               2  LIST OWNER INFORMATION                        *
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM ADDOWNER SCHLUM                                        -
              FNAME('NORBERT')                              -
              SNAME('SCHLUMBERGER')                         -
              DEPARTEMENT('International Projects (6304)')  -
              ADD1('IBM Deutschland Informationssysteme GmbH') -
              ADD2('Am Keltenwald 1')                       -
              ADD3('D-71139 Ehningen')                      -
              INTEL('919-3579')                             -
              EXTEL('++49-(0)-7034-15-3579')                -
              USER(SCHLUM)                                  -
              NODE(IBMDE)
/*
//STEP02   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM LISTOWNER SCHLUM
/*
```

The output of the TSO RMM LISTOWNER subcommand is shown in Example 3-29.

Example 3-29 Sample RMM LISTOWNER subcommand output

```
Owner       = SCHLUM
Last name   = SCHLUMBERGER           First names = NORBERT
Department  = INTERNATIONALE PROJECTS (6304)
Address     = IBM DEUTSCHLAND INFORMATIONSSYSTEME GMBH
              AM KELTENWALD 1
              71139 EHNINGEN (GERMANY)
Telephone:
Internal    = 919-3579               External    = +49-(0)7034-15-3579
Electronic mail:
Userid      = SCHLUM                 Node        = IBMDE

Volumes     = 0
```

Archived

3.4.3 Add racks to DFSMSrmm

The ADDRACK subcommand defines shelf locations in the removable media library and storage locations. DFSMSrmm defines shelf spaces in the removable media library as rack numbers and bin numbers in storage locations. Use the ADDRACK subcommand to define empty, or available, rack and bin numbers to DFSMSrmm. You must supply an initial rack number.

The SEARCHRACK subcommand creates a list of shelf locations defined in the removable media library. Shelf locations in the removable media library are called rack numbers.

Example:

Add 20 racks to the removable media library, starting with rack number *SC0000*. In STEP02 create a list of the created rack numbers (see Example 3-30).

Example 3-30 Sample JCL to add racks

```
//ADDRACK  JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ******************************************************************* *
//*          STEP 1  ADDING 20 RACKS                                    *
//*               2  DISPLAY INFORMATION                                *
//* ******************************************************************* *
//STEP01   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM ADDRACK SC0000 -
             LOCATION(SHELF) -
             MEDIANAME(3590) -
             COUNT(20)
/*
//STEP02   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM SEARCHRACK RACK(SC*) LIMIT(*)
/*
```

The output of the TSO RMM SEARCHRACK subcommand is shown in Example 3-31.

Example 3-31 Sample RMM SEARCHRACK subcommand output

```
Rack    Medianame  Volume  Status    Location
------  ---------  ------  --------  --------
SC0000  3590       SC0000  EMPTY     SHELF
SC0001  3590       SC0001  EMPTY     SHELF
```

Archived

```
SC0002  3590       SC0002  EMPTY     SHELF
SC0003  3590       SC0003  EMPTY     SHELF
SC0004  3590       SC0004  EMPTY     SHELF
SC0005  3590       SC0005  EMPTY     SHELF
SC0006  3590       SC0006  EMPTY     SHELF
SC0007  3590       SC0007  EMPTY     SHELF
SC0008  3590       SC0008  EMPTY     SHELF
SC0009  3590       SC0009  EMPTY     SHELF
SC0010  3590       SC0010  EMPTY     SHELF
SC0011  3590       SC0011  EMPTY     SHELF
SC0012  3590       SC0012  EMPTY     SHELF
SC0013  3590       SC0013  EMPTY     SHELF
SC0014  3590       SC0014  EMPTY     SHELF
SC0015  3590       SC0015  EMPTY     SHELF
SC0016  3590       SC0016  EMPTY     SHELF
SC0017  3590       SC0017  EMPTY     SHELF
SC0018  3590       SC0018  EMPTY     SHELF
SC0019  3590       SC0019  EMPTY     SHELF
20         ENTRIES LISTED
```

3.4.4 Add volumes to DFSMSrmm

The ADDVOLUME subcommand adds one or more volumes to DFSMSrmm. You have to define all volumes you will use in the DFSMSrmm CDS.

DFSMSrmm monitors all tape volume mounts and automatically updates information about your pre-defined tape volumes when they are used.

The SEARCHVOLUME subcommand creates a list of volumes matching the search criteria you specify. You can display lists of volumes according to ownership, assigned date, status, movement, action, pool, or media name.

Example:

Add 10 new scratch volumes in STEP01 to the removable media library using a rack number that is the same as the VOLSER. The volumes will not be initialized. Secondly, add 10 new scratch volumes to the removable media manager into the racks, but in this case the volumes must be labeled before they can be used, as indicated by INITIALIZE(Y). Thirdly, create a list of all previous added volumes (see Example 3-32).

Example 3-32 Sample JCL to add volume information

```
//STEP01   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM ADDVOLUME SC0000                       -
               STATUS(SCRATCH)              -
```

Archived

```
                INITIALIZE(N)               -
                LOCATION(SHELF)             -
                MEDIANAME(3590)             -
                MEDIATYPE(*)                -
                POOL(SC*)                   -
                COUNT(10)
 RMM ADDVOLUME SC0010                       -
                STATUS(SCRATCH)             -
                INITIALIZE(Y)               -
                LOCATION(SHELF)             -
                MEDIANAME(3590)             -
                MEDIATYPE(*)                -
                POOL(SC*)                   -
                COUNT(10)
 RMM SEARCHVOLUME VOLUME(SC*) OWNER(*) LIMIT(*)
/*
```

The output of the TSO RMM SEARCHVOLUME subcommand is shown in Example 3-33.

Example 3-33 Sample RMM SEARCHVOLUME subcommand output

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                      date       date
------ -------- ------ ---------- ---------- -------- ----- -- ------------
SC0000          SC0000 2001/133               SHELF    0     S
SC0001          SC0001 2001/133               SHELF    0     S
SC0002          SC0002 2001/133               SHELF    0     S
SC0003          SC0003 2001/133               SHELF    0     S
SC0004          SC0004 2001/133               SHELF    0     S
SC0005          SC0005 2001/133               SHELF    0     S
SC0006          SC0006 2001/133               SHELF    0     S
SC0007          SC0007 2001/133               SHELF    0     S
SC0008          SC0008 2001/133               SHELF    0     S
SC0009          SC0009 2001/133               SHELF    0     S
SC0010          SC0010 2001/133               SHELF    0     S
SC0011          SC0011 2001/133               SHELF    0     S
SC0012          SC0012 2001/133               SHELF    0     S
SC0013          SC0013 2001/133               SHELF    0     S
SC0014          SC0014 2001/133               SHELF    0     S
SC0015          SC0015 2001/133               SHELF    0     S
SC0016          SC0016 2001/133               SHELF    0     S
SC0017          SC0017 2001/133               SHELF    0     S
SC0018          SC0018 2001/133               SHELF    0     S
SC0019          SC0019 2001/133               SHELF    0     S
20         ENTRIES LISTED
```

Archived

3.4.5 Add bins to DFSMSrmm

The ADDBIN subcommand defines shelf locations in the storage locations. DFSMSrmm defines shelf spaces in storage locations as bin numbers. Use the ADDBIN subcommand to define empty bin numbers to DFSMSrmm.

The SEARCHBIN subcommand creates a list of shelf locations defined in the storage locations. Shelf locations in the storage locations are called bin numbers.

Example:

Add 20 empty shelf locations to the built-in storage location SANJOSE, starting with bin number 000001 (see Example 3-34). The example assumes that you have defined storage location SANJOSE with the LOCDEF command in PARMLIB member EDGRMMxx. STEP02 creates a list of the new bin numbers.

Example 3-34 Sample JCL to add bins

```
//ADDBIN   JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************** *
//*     STEP 1  ADDING 10 BINS TO built_in storage location SANJOSE    *
//*       2  DISPLAY INFORMATION                                       *
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM ADDBIN 000001 -
            LOCATION(SANJOSE) -
            MEDIANAME(3590) -
            COUNT(10)
/*
//STEP02   EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM SEARCHBIN BIN(*) LIMIT(*) LOCATION(SANJOSE)
/*
```

The result of the TSO RMM SEARCHBIN subcommand is shown in Example 3-35.

Example 3-35 Sample RMM SEARCHBIN subcommand output

```
Rack/Bin           Media              Current  Moving-in  Moving-out Old
number   Location  name      Status   volume   volume     volume     volume
-------- --------  --------  -------  -------  ---------  ---------- ------
000001   SANJOSE   3590      EMPTY
000002   SANJOSE   3590      EMPTY
```

```
000003   SANJOSE   3590      EMPTY
000004   SANJOSE   3590      EMPTY
000005   SANJOSE   3590      EMPTY
000006   SANJOSE   3590      EMPTY
000007   SANJOSE   3590      EMPTY
000008   SANJOSE   3590      EMPTY
000009   SANJOSE   3590      EMPTY
000010   SANJOSE   3590      EMPTY
SEARCH COMPLETE - MORE ENTRIES MAY EXIST
10        ENTRIES LISTED
```

3.4.6 Add a VRS to DFSMSrmm

The RMM ADDVRS command adds a new VRS to DFSMSrmm. A VRS is used to define retention and movement policies for data set and volumes.

The LISTVRS subcommand displays details about a single VRS. Specify a data set name when requesting information about a data set VRS.

Example:

Add a data set VRS to retain all data sets matching the DSNAME mask SCHLUM.RMMTEST.MOVE.** in location REMOTE until the data set is no longer cataloged (see Example 3-36). STEP02 requests information about the data set VRS defined for SCHLUM.RMMTEST.MOVE.** data sets.

Example 3-36 Sample JCL to add VRS

```
//ADDVRS   JOB ,14O.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ******************************************************************* *
//*          STEP 1  ADDING A VITAL RECORD SPECIFICATION                *
//*                  2  DISPLAY INFORMATION                             *
//* ******************************************************************* *
//STEPO1   EXEC PGM=IKJEFTO1
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM ADDVRS DSNAME('SCHLUM.RMMTEST.MOVE.**')         -
            CYCLES                                   -
            COUNT(99999)                             -
            DESCRIPTION('RETAIN AND MOVE DATA SETS') -
            LOCATION(REMOTE)                         -
            OWNER(SCHLUM)                            -
            STORENUMBER(1)                           -
            WHILECATALOG
/*
//STEPO2   EXEC PGM=IKJEFTO1
//SYSTSPRT DD   SYSOUT=*
```

Archived

```
//SYSTSIN  DD   *
 RMM LISTVRS DSNAME('SCHLUM.RMMTEST.MOVE.**')
/*
```

The result of the TSO RMM LISTVRS subcommand is shown in Example 3-37.

Example 3-37 Sample RMM LISTVRS subcommand output

```
Data set mask = SCHLUM.RMMTEST.MOVE.**                      Type = DSNAME
Job name mask =                              Retain until expired = NO
Count        = 99999 CYCLES                Retain while cataloged = YES
Delay        = 0     Days in the HOME location
Store number = 1     CYCLES  in the REMOTE   location
Priority     = 0                                  Release Options:
                                             Expiry date ignore = NO
                                              Scratch immediate = NO
Next VRS in chain    =          using     VRS

VRS Owner    = SCHLUM
Description  = RETAIN AND MOVE DATA SETS
Vital Record Specification to be deleted on 1999/365
```

3.4.7 Create tape data sets

Use the IEBGENER utility to create a multi-file, multi-volume chain to understand the differences between DFSMSrmm and TLMS. Example 3-38 shows sample JCL for creating different data sets on tape.

Example 3-38 Sample JCL to create data sets on tape

```
//CREATE1  JOB DE06304,14O.SCHLUMBERGER,....
//*       ************************************************************
//*       * 1. STEP   ALLOC 1. FILE  DSN=SCHLUM.RMMTEST.FILE1.VOL1   *
//*       *           VOLUME 1 / FILE 1 / RETPD 548O (APPROX. 15 Y.) *
//*       ************************************************************
//STEPO1   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=DFRMM.TEST.DATA.SMALL,DISP=SHR
//SYSUT2   DD   DSN=SCHLUM.RMMTEST.FILE1.VOL1,
//         DISP=(,CATLG,DELETE),
//         UNIT=IBMATL1,VOL=(,RETAIN),
//         DCB=*.SYSUT1,LABEL=RETPD=5480
//SYSIN    DD   DUMMY
//*       ************************************************************
//*       * 2. STEP   ALLOC 2. FILE  DSN=SCHLUM.RMMTEST.FILE2.VOL1+2 *
//*       *           VOLUME 1 FILE 2 + VOLUME 2 FILE 1 / RETPD = 5  *
//*       ************************************************************
```

```
//STEP02   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=DFRMM.TEST.DATA.BIG,DISP=SHR
//SYSUT2   DD   DSN=SCHLUM.RMMTEST.FILE2.VOL12,
//         DISP=(,CATLG,DELETE),
//         DCB=*.SYSUT1,LABEL=(2,SL,,RETPD=5),
//         VOL=(,RETAIN,,REF=*.STEP01.SYSUT2)
//SYSIN    DD   DUMMY *
//*        ************************************************************
//*        * 3. STEP   ALLOC 3. FILE  DSN=SCHLUM.RMMTEST.FILE3.VOL2   *
//*        *           VOLUME 2 / FILE 2 / RETPD 3                    *
//*        ************************************************************
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=DFRMM.TEST.DATA.SMALL,DISP=SHR
//SYSUT2   DD   DSN=SCHLUM.RMMTEST.FILE3.VOL2,
//         DISP=(,CATLG,DELETE),
//         DCB=*.SYSUT1,LABEL=(3,SL,,RETPD=3),
//         VOL=(,RETAIN,,REF=*.STEP02.SYSUT2)
//SYSIN    DD   DUMMY
//*        ************************************************************
//*        * 4. STEP   ALLOC 4. FILE  DSN=SCHLUM.RMMTEST.FILE4.VOL23  *
//*        *           VOLUME 2 FILE 3 + VOLUME 3 FILE 1 / RETPD = 4  *
//*        ************************************************************
//STEP04   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=DFRMM.TEST.DATA.BIG,DISP=SHR
//SYSUT2   DD   DSN=SCHLUM.RMMTEST.FILE4.VOL23,
//         DISP=(,CATLG,DELETE),
//         DCB=*.SYSUT1,LABEL=(4,SL,,RETPD=4),
//         VOL=(,RETAIN,,REF=*.STEP03.SYSUT2)
//SYSIN    DD   DUMMY
//*        ************************************************************
//*        * 5. STEP   ALLOC 5. FILE  DSN=SCHLUM.RMMTEST.FILE5.VOL3   *
//*        *           VOLUME 3 / FILE 2 / RETPD 10                   *
//*        ************************************************************
//STEP05   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=DFRMM.TEST.DATA.SMALL,DISP=SHR
//SYSUT2   DD   DSN=SCHLUM.RMMTEST.FILE5.VOL3,DISP=(,CATLG,DELETE),
//         DISP=(,CATLG,DELETE),
//         DCB=*.SYSUT1,LABEL=(5,SL,,RETPD=10),
//         VOL=(,RETAIN,,REF=*.STEP04.SYSUT2)
//SYSIN    DD   DUMMY
```

Archived

3.4.8 Allocate inventory management data sets

Before you can use the EDGHSKP utility to run inventory management activities, you must define several data sets, but not all data sets are required for all functions. Some data sets used during inventory management must be pre-allocated and cataloged because these data sets are used by both the EDGHSKP utility and the DFSMSrmm subsystem. Refer to the manual *DFSMSrmm Implementation and Customization Guide*, SC26-7405 for additional information and sample jobs. Example 3-39 shows sample JCL we used for allocating these data sets.

Example 3-39 Sample JCL to pre-allocate EDGHSKP data sets

```
//ALLOCATE JOB ,14O.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************** *
//*          STEPO1  pre-allocate MESSAGE data set                 *
//*                  pre-allocate REPORT data set                  *
//*                  pre-allocate RPTEXT data set                  *
//*                  pre-allocate XRPTEXT data set                 *
//*                  pre-allocate ACTIVITY data set                *
//* ************************************************************** *
//STEPO1   EXEC PGM=IEFBR14
//MESSAGE  DD DSN=RMM.PROD.MSGS,DISP=(NEW,CATLG),
//         UNIT=SYSDA,AVGREC=U,SPACE=(4O96,(1O,1O))
//REPORT   DD DSN=RMM.HSKP.REPORT.FILE.NAME,DISP=(NEW,CATLG),
//         LRECL=137,RECFM=VBA,
//         UNIT=SYSDA,AVGREC=U,SPACE=(4O96,(50,30))
//ACTIVITY DD DSN=RMM.HSKP.ACTIVITY.FILE.NAME,DISP=(NEW,CATLG),
//         UNIT=SYSDA,AVGREC=U,SPACE=(4096,(50,30))
//REPTEXT  DD DSN=RMM.HSKP.REPORT.EXTRACT.FILE.NAME,DISP=(NEW,CATLG),
//         UNIT=SYSDA,AVGREC=U,SPACE=(4096,(1000,500))
//XREPTEXT DD DSN=RMM.HSKP.REPORT.XEXTRACT.FILE.NAME,DISP=(NEW,CATLG),
//         UNIT=SYSDA,AVGREC=U,SPACE=(4096,(1000,500))
```

The REPTEXT and XREPTEXT DD statements allocate data sets that contain the extract copy of the DFSMSrmm control data set. The extract copy is called the extract data set. You must specify either the REPTEXT DD statement or the XREPTEXT DD statement when you use the EDGHSKP RPTEXT parameter. You do not need both data sets created. A single REPTEXT data set can be used for either normal extract or extended extract.

You can use also the sample JCL in SAMPLIB member EDGJHKPA to allocate ACTIVITY, MESSAGE, REPORT, and RPTEXT files required to run inventory management functions.

3.4.9 Run inventory management

After you have created racks, volumes, VRSs, and data sets, you must run inventory management as follows:

1. Run vital record processing (VRSEL) before expiration and storage location management processing to identify which volumes to retain and where volumes should be moved according to their VRSs.
2. Run storage location management (DSTORE) to set a destination location, and optionally assign shelf locations in storage locations to volumes that are being sent out of the removable media library.
3. Run expiration processing (EXPROC) to identify those volumes not required for vital records that are ready to expire. During expiration processing, release actions for volumes are noted.
4. Run extract data set processing (RPTEXT) to a sequential data set that contains information from DFSMSrmm and use it as input to EDGRPTD, the DFSMSrmm reporting utility.
5. Back up the CDS and optionally the journal (BACKUP), and reset the journal data set when the CDS and journal are backed up using the EDGHSKP utility.

Example 3-40 shows sample JCL to perform inventory management and to copy the message data set to your job log.

Example 3-40 Sample JCL to run inventory management

```
//EDGHSKP  JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ****************************************************************** *
//*  Step 01   List all volumes to be returned to scratch status       *
//*       02   Performing inventory management - EDGHSKP               *
//*            a) VRSEL  (Vital Record Processing)                     *
//*            b) DSTORE (Stores management processing)                *
//*            b) EXPROC (Expiration processing)                       *
//*            d) RPTEXT (Report extract processing)                   *
//*            e) BACKUP (Backing up the control data set, backing up  *
//*               the journal data set and clearing out the journal    *
//*               data set only if the back up of the control data set *
//*               was successfully)                                    *
//*       03   Copy DFSMSrmm message file to sysout                    *
//* ****************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 RMM SEARCHVOLUME VOLUME(*) ACTION(SCRATCH) LIMIT(*) OWNER(*)
/*
//STEP02   EXEC PGM=EDGHSKP,
```

```
//         PARM='VRSEL,DSTORE,EXPROC,RPTEXT,BACKUP,DATEFORM(J)'
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DSN=RMM.PROD.MSGS,DISP=SHR
//REPORT   DD   DSN=RMM.HSKP.VRSEL,DISP=SHR
//ACTIVITY DD   DSN=RMM.HSKP.ACTIVITY,DISP=SHR
//REPTEXT  DD   DSN=RMM.HSKP.EXTRACT,DISP=SHR
//BACKUP   DD   DSN=RMM.HSKP.BACKUP,DISP=(,CATLG,DELETE),
//         SPACE=(TRK,(150,10),RLSE),
//         DCB=(GDG,LRECL=9000,BLKSIZE=23476,RECFM=VB)
//JRNLBKUP DD   DSN=RMM.HSKP.JRNLBKUP,DISP=(,CATLG,DELETE),
//         SPACE=(TRK,(10,10),RLSE),
//         DCB=(GDG,LRECL=9000,BLKSIZE=23476,RECFM=VB)
//*        ************************************************************
//STEP03   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=0
//SYSUT1   DD   DSN=RMM.PROD.MSGS,DISP=SHR
//SYSUT2   DD   SYSOUT=*
//SYSIN    DD   DUMMY
//*        ************************************************************
```

3.4.10 Confirm movements

Confirm your outstanding volume moves by using the RMM CHANGEVOLUME command. In DFSMSrmm you have to confirm each volume move, except for volumes moved to an IBM tape library, to confirm that you have moved the volume.

You can confirm a move in two ways:

- To confirm a move for a single volume, use the CHANGEVOLUME subcommand with a VOLSER and the CONFIRMMOVE operand (see Figure 3-42).

```
RMM CHANGEVOLUME vvvvvv CMOVE
```

Figure 3-42 Confirm move for a single volume

No further actions are required.

- To perform a global confirm of volume movements, specify an asterisk for the VOLSER (see Figure 3-43).

```
RMM CHANGEVOLUME * CMOVE(ALL,SANJOSE)
```

Figure 3-43 Confirm move for all volumes

Archived

In this case you must rerun the storage location management processing to update the CDS.

3.4.11 Create reports

You can create several types of reports using the DFSMSrmm report utilities, EDGRPTD and EDGAUD, and also using the DFSMSrmm report generator. DFSMSrmm provides standard reports and samples shipped in SAMPLIB. You can use the sample DFSMSrmm EDGJRPT JCL that invokes the EDGRRPTE exec to create the reports. The input to the reporting execs is the extended extract data set. Documentation about the reports is shipped in SYS1.SAMPLIB in the EDGDOC file.

Use the reports created to help you keep track of your DFSMSrmm environment, and compare these reports with those of TLMS running in parallel. You can use DFSORTS's ICETOOL to write customized reports, or you can use the DFSMSrmm TSO subcommands to create lists of information defined to DFSMSrmm.

Create reports with EDGRPTD

Use EDGRPTD to create inventory reports for auditing the physical contents of the installation media library and storage locations. You can also use EDGRPTD to create movement reports that list volumes to be moved from one location to another. Use these reports to make an inventory of your volumes and to identify volumes that need to be pulled and moved to other locations.

DFSMSrmm-supplied reports

DFSMSrmm provides REXX execs and JCL in SYS1.SAMPLIB member EDGJRPT that you can use as-is to create the following reports:

- Pull list for scratch tapes sorted by volume serial number
- Pull list for scratch tapes sorted by data set name
- List of all volumes in scratch status
- List of volumes that are ready to scratch
- Ready to scratch
- Inventory list by volume serial number
- Inventory list by data set name
- Inventory of data set including number of KB used
- Inventory of volume serial number by location
- Inventory of data set names by location
- Inventory of BIN numbers by location
- List of all data set names at loan location
- List of all volume serial numbers at loan location
- List of all multiple volume, multiple data set

Archived

- Movement report including the first data set name on the volume
- Movement report sorted by storage location bin number
- Movement report sorted by volume serial number
- Inventory list sorted by volume serial number including volume count

You can copy these reports and use them to create reports that are tailored for your installation.

Create reports with EDGAUD

Use EDGAUD to create security and audit reports using either previously selected and sorted SMF records or raw SMF data. DFSMSrmm produces SMF records when you specify the SMFAUD or SMFSEC installation options in the PARMLIB member EDGRMMxx. DFSMSrmm uses the current subsystem startup option for the SMF record types and default report option unless you override them with the EDGAUD EXEC parameters.

Create reports with the report generator

The report generator guides you through ISPF panels to define your reports and then it produces JCL. The generated JCL includes all the commands needed for producing the report as you specified. For obtaining the report, you must submit the JCL. You can use this JCL for subsequent reports.

For detailed information on how to create a report, refer to 12.4, “DFSMSrmm report generator” on page 540.

3.4.12 Run EDGINERS to initialize volumes automatically

With EDGINERS, volumes can be initialized and erased, based on the volume status and actions recorded in the CDS, or you can let your users or operator perform the processing. Refer to *DFSMSrmm Guide and Reference*, SC26-7404, for more information about operator tasks such as responding to initialization messages, tape mount messages, and using the LABEL procedure to request EDGINERS processing, and to *DFSMSrmm Implementation and Customization Guide*, SC26-7405, for a description of the EDGLABEL procedure provided by DFSMSrmm.

During automatic processing, the process requests each volume to be mounted, as with any batch job, because an operator reply is not required. When the volume is mounted, processing continues. If the operator cannot mount a volume, DFSMSrmm allows them to skip processing the current requested volume by replying to a WTOR.

As part of volume expiration processing, DFSMSrmm records when volumes need to be initialized or erased as they are released for return to scratch. If you use the EXEC parameters in your JCL to set up automatic processing, DFSMSrmm initializes or erases those volumes without librarian intervention. EDGINERS issues WTOR messages and MSGDISP requests to the operator and drive to get a volume mounted and demounted.

You can run EDGINERS automatic processing to initialize the volumes you are adding to DFSMSrmm. When you add scratch volumes to DFSMSrmm, use the RMM TSO subcommand operand, INIT(Y), on the ADDVOLUME command to ensure that volumes are initialized before they are available as scratch. Specify the initialization parameters on the PARM statement in your EDGINERS JCL. DFSMSrmm selects all volumes marked as requiring initialization and performs automatic processing. Example 3-41 shows sample JCL for automatic processing.

Example 3-41 Sample JCL for EDGINERS automatic processing

```
//LCLHSKP  JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************* *
//* Step  01   Initializing volumes                               *
//* ************************************************************* *
//STEP01   EXEC PGM=EDGINERS,PARM='MEDIANAME(3590),VERIFY'
//SYSPRINT DD   SYSOUT=*
//TAPE     DD   UNIT=(TAPE,,DEFER)
```

This job initializes the volumes SC0010 through SC0019 added in Example 3-32 on page 116. The VERIFY parameter requests 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 read to ensure that operator errors have not occurred (for example, a mismatch between the internal label and the external label).

3.4.13 Restore the CDS

To restore the CDS, you first delete the existing CDS and then redefine it using Access Method Services (AMS) commands. The EDGBKUP utility is used to forward recover the CDS. During the recovery you must stop all DFSMSrmm activities using the MVS command in Figure 3-44.

```
MODIFY DFRMM,QUIESCE
```

Figure 3-44 QUISCE all DFSMSrmm activities

When the CDS recovery is complete, you must restart DFSMSrmm using the MVS operator command shown in Figure 3-45.

```
MODIFY DFRMM,M=xx
```

Figure 3-45 Restart DFSMSrmm

Using the backups of the CDS and journal taken by the EDGHSKP utility during inventory management, you can recover the DFSMSrmm CDS at any time. Figure 3-46 shows you how you can use the latest backup of the CDS and the current journal file to restore and forward recover to the point of failure.

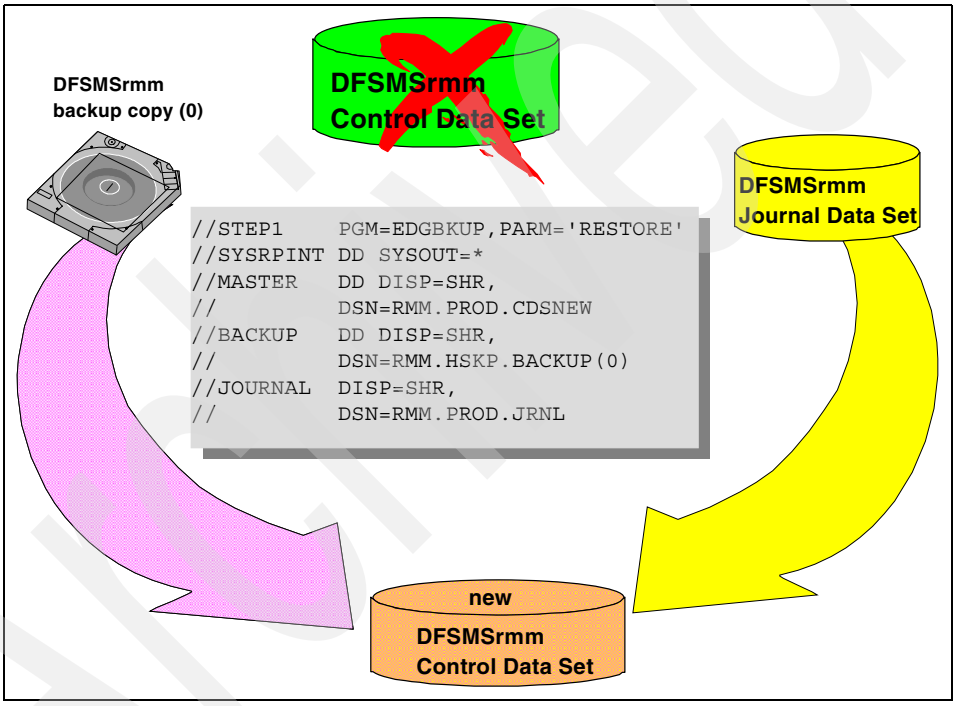


Figure 3-46 CDS recovering from latest backup

If the latest backup of the CDS is not available, you can use an older backup and the backups of the journal to recover. Figure 3-47 shows you how you can use the previous backup generation of the CDS, the latest journal backup, and the current journal file to restore and forward recover to the point of failure.

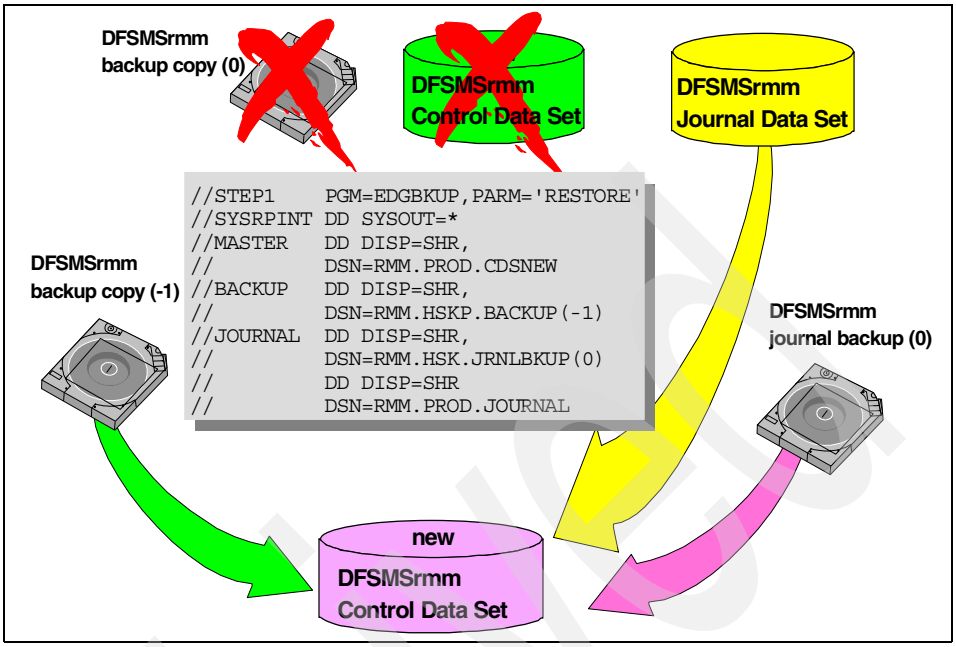


Figure 3-47 CDS recovering from backup-1

Using this process you can recover from any CDS backup as long as you have the corresponding journal backups.

You can restore the DFSMSrmm CDS and forward recover the restored CDS with the journal. Use the JOURNAL DD statement to concatenate multiple journal data sets. If you forward recover using multiple journal data sets, ensure that the journal data sets are concatenated in the correct order in which changes were originally made.

When a backup is taken using DFSMSdss, by specifying BACKUP(DSS) for any backup being restored, a journal backup is also required for forward recovery. For the latest CDS backup, both the latest journal backup and the active journal must be used for forward recovery. In a remote recovery site, the journal backup is all that is required for successful forward recovery of the restored CDS.

In our example we are using a different data set name for the DFSMSrmm control data set than the original one. If you continue with the new restored CDS, you have to rename it or you must update the EDGRMMxx PARMLIB member to specify the new name in the DSNAME option.

Example 3-42 shows sample JCL to delete and define the CDS before the forward recovery.

Example 3-42 Sample JCL to delete and define the CDS

```
//RESTORE  JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ************************************************************* *
//* Step  01   DELETE THE OLD CONTROL DATA SET                    *
//*       02   ALLOCATE A NEW CONTROL DATA SET                    *
//*       03   RESTORE THE CONTROL DATA SET                       *
//* ************************************************************* *
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.PROD.CDSNEW CL
  SET MAXCC=0
/*
//STEP02   EXEC PGM=IDCAMS                 ALLOCATION MASTER FILE
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,UNIT=3390,VOL=SER=DFRMM4
//SYSIN    DD   *
  DEFINE CLUSTER (NAME(RMM.PROD.CDSNEW)              -
                  FILE(MASTER)                       -
                  FREESPACE(15 0)                    -
                  KEY(56 0)                          -
                  REUSE                              -
                  RECSZ(512 9216)                    -
                  SHR(3 3)                           -
                  STORCLAS(storage_class)            -
                  VOLUMES(DFRMM4))                   -
         DATA     (NAME(RMM.PROD.CDSNEW.DATA)        -
                  CYLINDERS(300 300)                 -
                  CISZ(18432)                        -
                  FILE(MASTER))                      -
         INDEX    (NAME(RMM.PROD.CDSNEW.INDEX)       -
                  CYLINDERS(30 30)                   -
                  FILE(MASTER))
/*
```

Example 3-43 shows sample JCL to forward recover the CDS for a backup with BACKUP(AMS). In this case you have to specify a JOURNAL DD statement with the actual journal.

Example 3-43 Sample JCL to forward recover the CDS (backup with AMS)

```
//STEP03   EXEC PGM=EDGBKUP,PARM='RESTORE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DSN=RMM.PROD.CDSNEW,DISP=SHR
//BACKUP   DD   DSN=RMM.HSKP.BACKUP(0),DISP=SHR
//JOURNAL  DD   DSN=RMM.PROD.JRNL,DISP=SHR
```

Archived

Example 3-44 shows sample JCL to forward recover the CDS for a backup with BACKUP(DSS).

Example 3-44 Sample JCL to forward recover the CDS (backup with DSS)

```
//STEP03   EXEC PGM=EDGBKUP,PARM='RESTORE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DSN=RMM.PROD.CDSNEW,DISP=SHR
//BACKUP   DD   DSN=RMM.HSKP.BACKUP(0),DISP=SHR
//JOURNAL  DD   DSN=RMM.HSKP.JRNLBKUP(0),DISP=SHR
//         DD   DSN=RMM.PROD.JRNL,DISP=SHR
```

You can check that restore and forward recovery were successful by displaying volume information. All volumes should have been initialized if the job shown in Example 3-41 on page 127 ended without any errors.

3.4.14 Verify the CDS

Use the EDGUTIL utility to verify the contents of the CDS. You should perform the verification each time you recover your DFSMSrmm CDS. Example 3-45 shows sample JCL to verify the CDS.

Example 3-45 Sample JCL to verify the CDS contents

```
//VERIFY   JOB ,140.SCHLUMBERGER,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1)
//* ******************************************************************* *
//* STEP  01   VERIFYING THE CONTENTS OF THE CONTROL DATA SET          *
//* ******************************************************************* *
//STEP01   EXEC PGM=EDGUTIL,PARM='VERIFY(ALL)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDSNEW
//SYSIN    DD   DUMMY
```

Note: If your specifying VERIFY(ALL) DFSMSrmm performs extra processing with the TCDB and library manager. That can be a very long running process. DFSMSrmm scans both the DFSMSrmm control data set and the TCDB sequentially to find DFSMSrmm volumes that are not in the TCDB, and TCDB volumes that are not in the DFSMSrmm control data set. DFSMSrmm also checks any volume that is found to be system-managed, either by definition to DFSMSrmm or retrieved from the TCDB, against the library manager database for an IBM automated tape library.

Archived

3.4.15 Testing implemented exits

If you use the following exits, you must test them:

- **EDGUX100**

  Use the DFSMSrmm EDGUX100 installation exit to:

  - Ignore foreign or duplicate volumes and optionally provide an external volume serial number for use in messages intercepted and updated by DFSMSrmm.
  - Use JCL special expiration dates to manage volumes by setting vital record specification management values to retain data sets.
  - Set any expiration date, including a zero value, for a data set.
  - Perform pooling management:
    - Manage scratch pools based on job name and data set name.
    - Select a pool to be used for a non-specific tape volume request.
    - Select a specific pool to be used for a non-specific tape volume request and request that the tape drive cartridge loader is disabled.
  - Pass vital record specification management value and pooling decisions to pre-ACS processing that can be used in ACS routines to assign storage group and management class.
  - Obtain information to modify input from DFSMSrmm disposition processing.
  - Request the recording of only the first file on a multi-file volume.
  - Change the location for a volume.
  - Support the use of the storage group name for pooling for volumes that reside in manual tape libraries.

  You can use the supplied sample EDGUX100 exit to support the special expiration dates 98000 and 99000 without modifications to the sample code. If you plan to support other special dates using the exit and have modified it, you also need to test them. The results and the testing of the additional dates are the same as for the 99000 date.

  To test use of the 99000 special date, create a new tape data set by specifying EXPDT=99000. Use the DFSMSrmm LISTDATASET command or the list data set dialog to display details of the data set that should include D99000 as the VRS management value.

Archive

To test the use of EXPDT=98000, first define a RACF FACILITY class resource (see Appendix C, “Security topics” on page 647) to allow the use of the ignore function to bypass DFSMSrmm recording. Whether you test to check that ignore is allowed or that ignore is not allowed, DFSMSrmm issues a message of explanation. To test the ignore function, DFSMSrmm must be running in either warning or protect mode.

- **EDGUX200**

  Use the DFSMSrmm EDGUX200 installation exit to:

  - Return a volume to scratch status in an external inventory
  - Prevent a volume from returning to scratch status
  - Request that DFSMSrmm ignores data set information recorded for a volume

  The sample EDGUX200 provides no specific function. Test the exit only if you plan to use it and have modified the supplied source code. The exit is intended for communication with software that has an inventory of scratch tapes. To test the exit, you need a tape volume that is in pending release status and is to return to scratch. Run inventory management EXPROC processing and then check that the external inventory has been correctly updated.

- **System-managed tape support by using the OAM exits**

  - CBRUXENT (cartridge entry exit)
  - CBRUXEJC (cartridge eject exit)
  - CBRUXCUA (cartridge change use attributes exit)
  - CBRUXVNL (volume not in library exit).

  Test these exits only if you have an IBM system managed tape library. DFSMSrmm provides sample exits that are ready to use, installed and active on your system, and do not require modification. However, you can customize the exits to meet your needs. CBRUXVNL is the exit you may want to consider customizing to meet your needs.

  A simple set of tests can be used to test these exits:

  a. Define a volume to DFSMSrmm for entry to the library as shown in Figure 3-48.

```
RMM ADDVOLUME volser LOCATION(ATL) STATUS(MASTER)
```

Figure 3-48 RMM ADDVOLUME subcommand with location ATL

  If you list or display the volume, it should be in MASTER status with a destination of ATL.

  b. Enter the volume into the library.

i. To test CBRUXENT, list or display the volume in DFSMSrmm. It should now show a location of ATL and no destination. If you display the volume using ISMF, the volume should display as private.

ii. To test CBRUXCUA, use ISMF to change the volume status to scratch. If DFSMSrmm is in protect mode, the change should be rejected. If DFSMSrmm is in warning mode, the change should be allowed, but a warning message should be issued that DFSMSrmm only allows the change in warning mode. Messages are not issued in record or manual mode.

iii. To test CBRUXEJC, eject the volume from the library. You can use any method for the ejection. When the eject is completed and the volume displays in DFSMSrmm, the volume should show as being in transit.

iv. To test CBRUXVNL, ensure that the volume is in private or master status, and run a job to create or read a file on the tape.

You should see message EDG8124I on the console that lists DFSMSrmm information or message EDG8123D. After receiving EDG8124I, depending on the circumstances, either WTOR EDG8121D or EDG8122D appear on the console. Reply RETRY after the volume is entered into the library, and the job should run successfully.

To check that all the OAM exits are enabled, you can use the MVS command shown in Figure 3-49.

```
D SMS,OAM
   F OAM,D,OAM,L=SCHLUM-Z
   CBR1100I OAM status: 351
   TAPE TOT  ONL  TOT  TOT  TOT  TOT   TOT   ONL   AVL   TOTAL
        LIB  LIB  AL   VL   VCL  ML    DRV   DRV   DRV  SCRTCH
          2    2    1    1    0    0    72     6     6    1392
   There are also 0 VTS distributed libraries defined.
   CBRUXCUA processing ENABLED.
   CBRUXEJC processing ENABLED.
   CBRUXENT processing ENABLED.
   CBRUXVNL processing ENABLED.
```

Figure 3-49 Display OAM status

3.4.16 Testing application use of tape

You should be familiar with the ways in which tape is used in your installation. Although we assume that everyone's use is the same, it is surprising how often we find new and devious uses of tape. To avoid discovering problems in production with DFSMSrmm, we suggest that you test the ways in which you use tapes today to ensure that DFSMSrmm supports them as expected. For example, you may use BLP (perhaps to scratch tapes) or non-labeled tapes, or you may have key business functions that depend on foreign tape processing.

Use tests that you have already in place for your production tape usage under DFSMSrmm to validate that they are supported and that you know which actions may be required to support them. For example, RACF profiles are required to use NL output to scratch and to process tapes that are to be ignored by DFSMSrmm.

NL scratch tapes

DFSMSrmm normally supports only standard label scratch tapes for non-specific NL tapes, and relabels them at time of use to NL and standard label at return to scratch. To use NL scratch tapes you can use the system tape exit IFG019VM. DFSMSrmm now ships a sample IFG019VM exit as member EDG019VM in SAMPLIB. The sample exit checks for a supported request and issued a WTOR to the operator to obtain the volume serial number. Testing should ensure that it does support the installation needs for non-specific NL tape.

Interface between DFSMShsm and DFSMSrmm

You should test the built-in interface between DFSMShsm and DFSMSrmm. Before you start this kind of test, ensure that DFSMShsm has the correct authority for the STGADMIN resources in RACF CLASS FACILITY. If you have a release of DFSMSrmm earlier than OS/390 Version 1 Release 4.0 you must define the use of the ARCTVEXT installation exit in the DFSMShsm PARMLIB member. You can test the function of the EDGTVEXT interface, the use of the ARCTVEXT exit EDGTVEXT interface, or the use of the ARCTVEXT exit by deleting a volume from the DFSMShsm tape volume pool using the HSM DELVOL command.

The TVEXTPURGE PARMLIB operand specifies how DFSMSrmm processes volumes to be purged by callers of EDGTVEXT or EDGDFHSM. It is the interface between DFSMShsm and DFSMSrmm.

There are three options that can be specified for TVEXTPURGE.

NONE DFSMSrmm takes no action for volumes to be purged.

RELEASE DFSMSrmm releases volumes to be purged according to the release actions set for the volume. You must run expiration processing to return a volume to scratch status.

EXPIRE Use the EXPIRE option to set the volume expiration date to the current date for volumes to be purged.

The recommended option is TVEXTPURGE(RELEASE).

3.5 Education

You should consider which groups of users may be affected by the initial introduction of DFSMSrmm for testing. Your colleagues, the operators, and possibly the tape librarian will need to learn about DFSMSrmm at some stage. At this stage a minimum level of education should be given so that any potential impact on your production system can be minimized by ensuring that affected users are aware of changes.

3.6 New operator procedures

When you start to run DFSMSrmm, system operators will most likely be affected. DFSMSrmm can cause the following changes:

- Mount and fetch messages are updated.
- Scratch pool names may be different and are included in the MVS mount messages.
- Tape drive displays are updated with rack number and pool information.
- DFSMSrmm issues WTORs that require operator action and decisions.

In addition, you may want the operator to be involved in starting and stopping DFSMSrmm.

Refer to *DFSMSrmm Guide and Reference*, SC26-7404, for details of messages and activities that affect the operator.

Archive

4

DFSMSrmm ACS support

Starting with DFSMSrmm V2.10, you can use your SMS ACS routines to assign a VRS management class and to select a specific scratch pool. In this chapter we describe the steps to update your SMS environment to allow a management class to be used in place of the VRS management value assigned in the EDGUX100 user exit, and a storage group name to be used as a scratch pool name.

Objectives of this chapter:

- To understand the interaction between DFSMSrmm and SMS
- Learn how DFSMSrmm calls SMS in both an SMS-managed environment and a non-SMS-managed environment
- Update the SMS configuration to move decisions from the EDGUX100 exit to SMS

Archived

© Copyright IBM Corp. 2004. All rights reserved.

4.1 DFSMSrmm SMS ACS support

With DFSMSrmm V2R10 or higher, you can use your SMS ACS routines to select a specific scratch pool and storage group to be assigned to create new data on tape, and set a management class to retain and move a data set on tape. You move your policy and scratch pool decisions to your SMS ACS routines as shown in Figure 4-1. For non-system managed tapes, DFSMSrmm calls the ACS routines to allow a managment class and storage group assignment. For SMS managed tapes, SMS calls the ACS routines directly.

To enable DFSMSrmm ACS support, you must have the SMS subsystem active and have a valid SMS configuration.

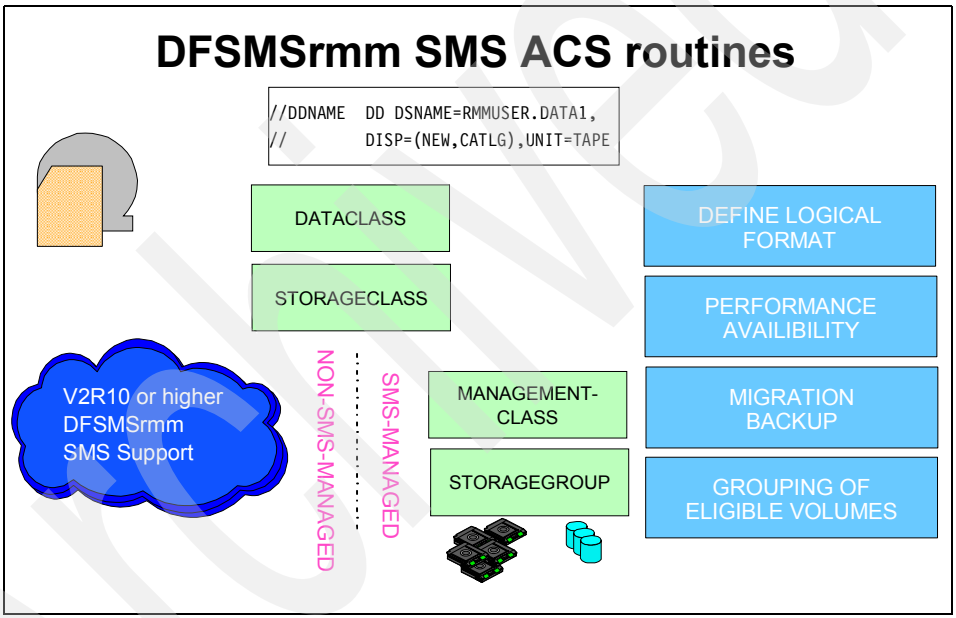


Figure 4-1 SMS ACS support with DFSMSrmm V2R10 or higher

With OS/390 V2R10 or higher, SMS ACS routines are always called by DFSMSrmm if you have a tape mount outside a SMS managed tape library.

ACS processing for non-system managed tapes can be tailored by the PARMLIB member EDGRMMxx OPTION subparameters PREACS and SMSACS. Figure 4-2 shows the syntax of the EDGRMMxx OPTION subparameter.

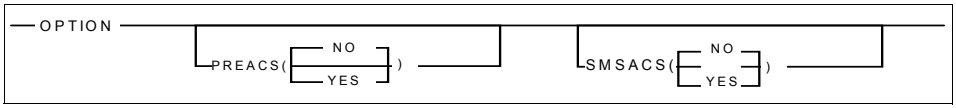


Figure 4-2 DFSMSrmm ACS processing PARMLIB options

PREACS

Specify the PREACS operand to control whether DFSMSrmm-supplied values and EDGUX100 installation exit-supplied values are input to SMS pre-ACS processing:

NO Specify NO to avoid DFSMSrmm pre-ACS processing using the DFSMSrmm EDGUX100 installation exit.

YES Specify YES to enable DFSMSrmm pre-ACS processing using the DFSMSrmm EDGUX100 installation exit.

SMSACS

The scratch pooling and policy assignment rules are per default coded in EDGUX100 user exit. You can move your decisions to the SMS ACS routines, where you can use ACS input variables as a base for the management class and storage group assignment:

NO Specify NO to prevent DFSMSrmm from calling the SMS ACS processing to obtain management class and storage group names. DFSMSrmm system-based scratch pooling, and VRS management values based on the EDGUX100 installation exit are used.

YES Specify YES to enable DFSMSrmm calls to the SMS ACS processing to obtain management class and storage group names. If values are returned by the SMS ACS routines, the values are used instead of the DFSMSrmm and EDGUX100 decisions.

We recommend that you move pooling decisions and VRS managment value assignment out of the EDGUX100 user exit.

Recommendation: Do not implement PREACS processing, and do not use the EDGUX100 user exit except for the following reasons:

- Cartridge loader control (ACL)
- Data set recording
- Sticky label customization
- Request ignore of volumes

Archived

4.1.1 How it works

Use ACS routines for scratch pooling based on tape storage group names. Using ACS processing to set a storage group name overrides all other pool selection methods. DFSMSrmm provides support for non-system-managed tape and for system-managed manual tape libraries. This support enables pooling at the individual volume level. You assign a storage group name to each volume by using DFSMSrmm TSO subcommands, or by using pooling information that you define with the DFSMSrmm EDGRMMxx PARMLIB VLPOOL command (see “Tailoring your EDGRMMxx PARMLIB member” on page 162). DFSMSrmm calls ACS routines passing environment information, including the pool identified by DFSMSrmm system-based pooling. The ACS routine can optionally set a storage group name, which overrides the DFSMSrmm system-based pool.

4.1.2 ACS support for non-system managed volumes

In a non-system-managed library, DFSMSrmm supports SMS ACS routines in different ways:

- DFSMSrmm will directly call the ACS routines if you have specified SMSACS(YES) in the EDGRMMxx PARMLIB member to allow a management class and storage group to be assigned. Figure 4-3 shows you how it works.

  Use ACS routines for scratch pooling based on tape storage group names. Using ACS processing to set a storage group name overrides all other pool selection methods. DFSMSrmm provides support for non-system-managed tape and for system-managed manual tape libraries. This support enables pooling at the individual volume level. You assign a storage group name to each volume by using DFSMSrmm TSO subcommands, or by using pooling information that you define with the DFSMSrmm EDGRMMxx PARMLIB VLPOOL command. DFSMSrmm calls ACS routines passing environment information (see 4.1.5, “Implementing SMS ACS processing” on page 149), including the pool identified by DFSMSrmm system-based pooling. The ACS routines can optionally set a storage group name, which overrides the DFSMSrmm system-based pool.

Archive

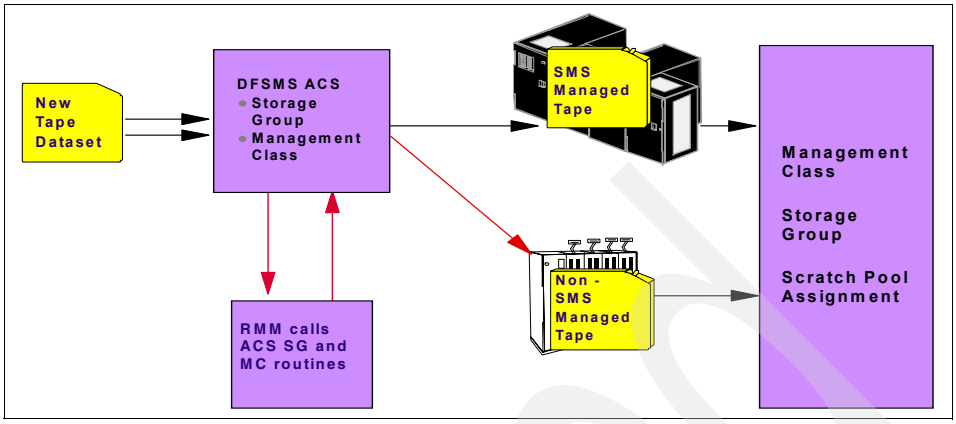


Figure 4-3 How DFSMSrmm SMS direct calls for non-system managed volumes

- You can use the EDGUX100 user exit to call SMS ACS routines to assign a VRS management class value and a scratch pool if you have specified PREACS(YES) as shown in Figure 4-4.

  Use pre-ACS processing to obtain the DFSMSrmm system-based pool or the EDGUX100 installation exit pool prefix as an input value to the ACS routines in the MSPOOL read-only variable. During pre-ACS processing, DFSMSrmm does not make the RMMPOOL environment call to the ACS routine. During pre-ACS processing for new allocations:

  – DFSMSrmm uses the VLPOOL definitions to select a default DFSMSrmm pool using DFSMSrmm system-based pooling. DFSMSrmm sets a pool prefix if a specific pool is selected.

  – DFSMSrmm calls the EDGUX100 installation exit to obtain a pool prefix value. If a pool prefix value is returned, the pool prefix value returned by the EDGUX100 installation exit overrides the DFSMSrmm selected pool.

  – DFSMSrmm returns the selected value in the MSPOOL read-only variable if the MSPOOL variable is not already set by the pre-ACS exit.

Archived

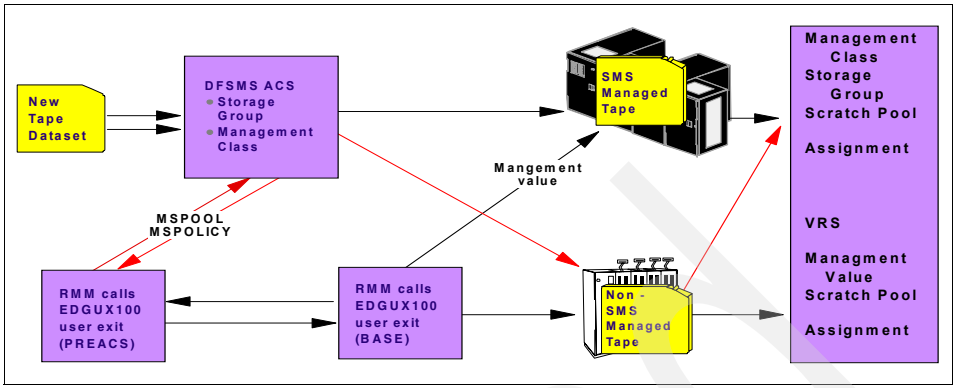


Figure 4-4 DFSMSrmm EDGUX100 user exit call with PREACS(YES)

- You can use both the DFSMSrmm direct call and the user exit EDGUX100 call together to take the decision as shown in Figure 4-5.

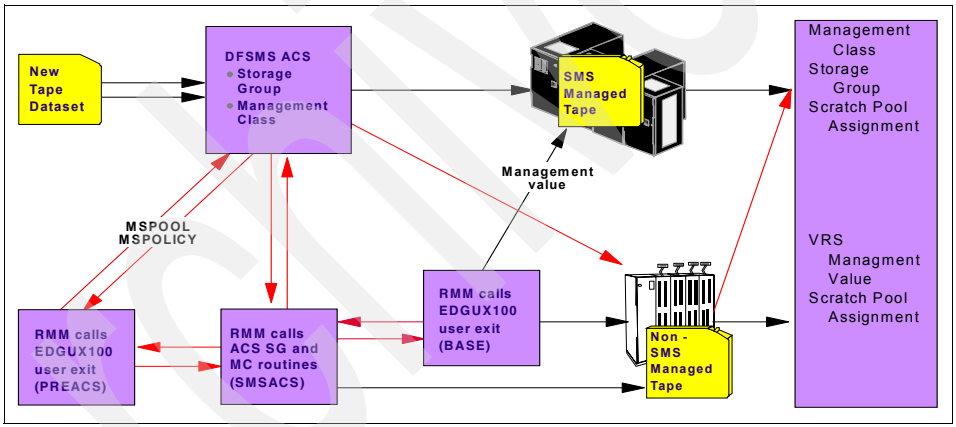


Figure 4-5 DFSMSrmm SMS direct and EDGUX100 user exit call

Note: To enable DFSMSrmm ACS support, you must have the SMS subsystem active and have a valid SMS configuration.

If you are using PREACS processing, and you have installed the EDGUX100 user exit, DFSMSrmm processes the functions in following sequence:

1. ACS processing calls the PRE-ACS exit.
2. DFSMSrmm calls EDGUX100 and sets MSPOOL and MSPOLICY.

3. ACS decides whether the entity is system managed or not. During processing, the four PRE-ACS read-only variables can be used as input to the decision.
4. For non-system managed tape, we enter DFSMSrmm ACS support. ACS processing is called for storage group processing, and again for management class processing.
5. DFSMSrmm calls EDGUX100 for its basic exit function.

4.1.3 Non-system managed tape libraries

A non-system managed tape library is all the volumes, shelves, and drives that are not in an automated tape library dataserver or manual tape library. You might know this as the traditional tape library in a data center, or as an automated environment that is not system-managed. DFSMSrmm provides complete tape management functions for the volumes and shelves in this traditional tape library.

All tape media and drives supported by z/OS are supported in this environment. Use DFSMSrmm to fully manage all types of tapes in a non-system managed tape library, including 3420 reels, 3480, 3490, 3590, and 3592 cartridge system tapes.

You can also use DFSMSrmm to manage volumes in any automated tape library that has special software including an IBM Tape Library dataserver that is managed using Basic Tape Library Support (BTLS).

Most non-IBM libraries do not provide support for system-managed tape, so we consider them to be the traditional tape libraries.

The IBM direction is to replace DFSMSrmm exit function with an alternative; either SMS policy or PARMLIB option.

4.1.4 SMS read-only variables

DFSMSrmm calls the ACS routines to request the assignment of storage group and management class names for non-system managed tape data sets. Figure 4-1 lists the read-only variables that are set for DFSMSrmm requests.

Table 4-1 SMS read-only variables set by DFSMSrmm

Variables			
&ACCT_JOB	&ACCT_STEP	&ACSENVIR	&DD
&DSN	&DSORG	&DSTYPE	&EXPDT
&FILENUM	&GROUP	&HLQ	&JOB

Variables			
&LABEL	&LIBNAME	&LLQ	&MSPOOL
&MSDEST	&MSPOLICY	&MSPARM	&NQUAL
&PGM	&STORGRP	&SYSNAME	&SYSPLEX
&UNIT	&USER	&XMODE	

Note: The SMS pre-ACS exit variables &MSPOOL, &MSPOLICY, &MSDEST, and &MSPARM read-only variables can be used by a tape management system only.

When the &ACSENVIR variable is set to RMMPOOL, DFSMSrmm requests SMS to return a storage group name. DFSMSrmm requests that both the management class and storage group ACS routines are run. With a combination of these routines, you can decide whether or not you want SMS to return a storage group value, and what the value will be. If a storage group name is returned, it must be a valid tape storage group name. Using different &ACSENVIR values helps you differentiate between DFSMSrmm requests for a storage group name, and allocation requests for system-managed data sets.

When the &ACSENVIR variable is set to RMMVRS, DFSMSrmm requests SMS to return a management class name. DFSMSrmm requests that only the management class ACS routine is run. Using different &ACSENVIR values helps you differentiate between DFSMSrmm requests for a management class name and allocation requests for system-managed data sets.

Use of the read-only variables

Most ACS variables are read-only. Read-only variables contain data set and system information, and they reflect what is known at the time of the allocation request. You can use read-only variables in comparison operations, but you cannot change their values.

Note: The read-only variables *are* case sensitive.

- **&ACCT_JOB**
  The accounting information from the JOB statement.
  Type: Literal
  Max value: 142 characters
- **&ACCT_STEP**
  The accounting information from the EXEC statement. This information is

refreshed for each step in the job.
Type: Literal
Max value: 142 characters

- **&ACSENVIR**
  The environment in which the ACS routine was invoked, one of:

RECALL	For data set recall operations
RECOVER	For data set recover operations
RENAME	For data set alter rename operations
RMMPOOL	For DFSMSrmm requests for a storage group name
RMMVRS	For DFSMSrmm requests for a management class name
CONVERT	For data set convert in place operations
ALLOC	For new data set allocations (this is the default)
STORE	OSREQ object store environment
CHANGE	OSREQ object change environment
CTRANS	OSMC object class transition environment
other	Installation exit can set its own value before re-invoking ACS

  Type: Literal
  Max value: 8 characters

- **&DD**
  DDNAME in the DD statement of the data set.
  Type: Literal
  Max value: 8 characters

- **&DSN**
  The name of the data set or collection for which ACS processing is taking place. For VSAM data sets, only the cluster name is passed to the ACS routine; the component names are not.
  If the data set has an absolute or relative generation number, it is stripped from &DSN. The generation number is the low-level qualifier of the data set name.
  Type: Literal
  Max value: 44 characters

- **&DSORG**
  The data set organization, one of:

PS	Physical sequential
PO	Partitioned
VS	VSAM organization
DA	BDAM organization
null	No value specified

  Type: Literal
  Max value: 2 characters

- **&DSTYPE**
  The data set type, one of:

Archive

GDS	One generation data set of a generation data group, or any data set allocated with a relative generation number (such as A.B.C(+1)) or an absolute generation number (such as A.B.C.G0000V00).
PERM	Standard permanent data sets
TEMP	Temporary data sets
null	None of the above

Type: Literal
Max value: 8 characters

- **&EXPDT**
  The expiration date in the form of YYYYDDD where YYYY is a year from 1900 - 2155 and DDD is a day in a year from 1 - 366.
  Exception: Expiration dates of 99365 and 99366 are considered "NEVER-SCRATCH" dates.
  Type: Literal
  Max value: 7 characters
- **&FILENUM**
  The value of the FILENUM ACS read-only variable. This variable corresponds to the data set sequence number on the JCL LABEL parameter. The default is 1. This field is optional.
  Type: Numeric
  Max value: 4 characters
- **&GROUP**
  The RACF-defined group to which you are connected, or the group specified in the GROUP keyword on the JCL JOB statement. If the environment is recall or recover, &GROUP is set only if the requester of the recall or recover is not a DFSMShsm authorized user. When DFSMShsm invokes the ACS routines, &GROUP is the group associated with &USER
  Type: Literal
  Max value: 8 characters
- **&HLQ**
  The high-level (first) qualifier of the data set or collection name.
  Type: Literal
  Max value: 8 characters
- **&JOB**
  The job name, the started task name, or the TSO/E userid from the JOB statement, depending on the execution mode (&XMODE).
  Type: Literal
  Max value: 8 characters
- **&LABEL**
  The value of the LABEL ACS read-only variable. This variable corresponds to the label field of the JCL LABEL parameter. Allowable values are NL, AL, SL,

Archived

NSL, SUL, AUL, BLP, LTM or blank. The default is IBM Standard Label. This field is optional.
Type: Literal
Max value: 3 characters

- **&LIBNAME**
  The name for the LIBNAME ACS read-only variable, can contain a 1 to 8 character tape library name. This field is optional.
  Type: Literal
  Max value: 8 characters
- **&LLQ**
  The low-level (last) qualifier of the data set or collection name.
  Type: Literal
  Max value: 8 characters
- **&MSPDEST**
  The destination, specified in data set name format, for a tape management system-driven tape allocation. This value is specified through the AMS pre-ACS installation exit. The data set name format lets you specify a sequence of destinations to be identified, where each qualifier is a specific destination. For example, a data set vaulted first at location OUTD and then sent to OLTS can have an MSPDEST of'OUTD.OLTS'. The actual values depend on the support provided by your tape management system.
  Type: Alphanumeric
  Max value: 44 characters
- **&MSPARM**
  Additional information related to a tape management system-driven tape allocation. This is a variable length field that can be indexed. The value is specified through an external exit.
  Type: Alphanumeric
  Max value: 256 bytes, including a 2–byte length field for each value specified
- **&MSPOLICY**
  The name of a management policy associated with tape data for a tape management system-driven allocation. You can use the DFSMSrmm EDGUX100 installation exit to set MSPOLICY to a VRS management value name. You can also set the value of this variable using the SMS pre-ACS installation exit, or allow your tape management system to set it using the pre-ACS installation exit.
  Type: Alphanumeric
  Max value: 8 characters
- **&MSPOOL**
  A tape pool name associated with the data set being allocated. In a system-managed tape environment with scratch pool support, you can use this variable to specify a default storage group, where the tape storage group

Archived

is equivalent to the tape pool specified in the variable. If you use the DFSMSrmm EDGUX100 installation exit, you can set this variable to the pool name or prefix determined by the DFSMSrmm scratch pool processing. This variable can also be set through the pre-ACS installation exit.
Type: Alphanumeric
Max value: 8 characters

- **&NQUAL**
  The number of qualifiers in the data set or collection name.
  Type: Numeric
  Max value: 22
- **&PGM**
  The name of the program the system is running.
  Type: Literal
  Max value: 8 characters
- **&SYSNAME**
  Specifies the system name of the system on which the ACS routine is executing. This field is optional.
  Type: Literal
  Max value: 8 characters
- **&SYSPLEX**
  Specifies the Parallel Sysplex name of the system on which the ACS routine is executing. This field is optional.
  Type: Literal
  Max value: 8 characters
- **&UNIT**
  IBM-supplied or installation defined generic name for a device type (for example, 3380, SYSDA).
  Type: Literal
  Max value: 8 characters
  A slash (/) preceding a four digit number represents a unit address (for example, /3380).
  When you allocate a tape data set with DISP=MOD, and no unit information is specified in the JCL, this variable is blank and SMS might attempt to manage the tape data set as a DASD-resident data set.
- **&USER**
  The user ID of the person allocating the data set. When DFSMShsm invokes the ACS routines, &USER is either the requestor of the recall or recover, or the user ID of the DFSMShsm address space. If the environment is recall or recover, the variable is set only if the requestor of the recall or recover is not a DFSMShsm authorized user.
  Type: Literal
  Max value: 8 characters

Archived

- **&XMODE**
  The execution mode in which the data set is being allocated, one of:

BATCH	Batch execution mode
TSO	TSO execution mode
TASK	A started address space

  Type: Literal
  Max value: 8 characters

4.1.5 Implementing SMS ACS processing

You can move your scratch pool and policy assignment decisions to the SMS ACS routines, where you can use ACS input variables as a base for management class and storage group assignment. In the EDGRMMxx PARMLIB member specify `SMSACS(YES)` so that DFSMSrmm calls the SMS ACS routines for pooling decisions and policy assignment.

Before you enable SMS ACS support in your installation, you must make some changes in your SMS ACS routines.

Define your SMS environment

To define a SMS environment for non-system managed tapes using the ISMF dialog as shown in Figure 4-6, following the steps in the same sequence as described below:

- “Define a tape library” on page 150
- “Defining a tape storage group” on page 153
- “Define management classes” on page 157
- “Tailoring your SMS management class routines” on page 159
- “Tailoring your SMS storage group routines” on page 161
- “Tailoring your EDGRMMxx PARMLIB member” on page 162
- “Adding volumes to the volume pool” on page 163
- “Defining your vital record specifications (VRSs)” on page 164

Archived

```
                ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R5
Enter Selection or Command ===> 10

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

Figure 4-6 ISMF primary option menu

Define a tape library

If you do not have such a library, or do not want to associate the SMS storage groups with an existing library, you must define a new library to ISMF. You can do this using the ISMF define library application, but you specify a made-up library ID. For other data fields required for tape library definition, you can select any values you would like. When you do this, OAM issues a `CBR3006I` message at startup time.

Note: You can skip this section if you do not make the storage group assignments.

1. Choose Option **10** (**Library Management**) on the ISMF PRIMARY OPTION MENU panel (see Figure 4-6) to display the LIBRARY MANAGEMENT SELECTION MENU.
2. Choose Option **3** (**Tape Library**) to display the TAPE LIBRARY APPLICATION SELECTION as shown in Figure 4-7.

```
                        LIBRARY MANAGEMENT SELECTION MENU
 Enter Selection or Command ===> 3

 1  Optical Library          - Optical Library Configuration
 2  Optical Drive            - Optical Drive Configuration
 3  Tape Library             - Tape Library Configuration
```

Figure 4-7 ISMF library selection menu

3. Define a system-managed tape library with a made-up library ID using the Tape Library Application Selection panel as shown in Figure 4-8.

```
                      TAPE LIBRARY APPLICATION SELECT
Command ===>

To Perform Library operations, Specify:

  CDS Name  . . . . . . . 'SYS1.SMS.SCDS'
                          (1 to 44 Character Data Set Name or 'Active')
  Library Name  . . . . . NOSMSLIB        (For Tape Library list, fully or
                                           Partially Specified or * for all)

Select one of the following options  :
  3  1. List    - Generate a list of Libraries
     2. Display - Display a Library
     3. Define  - Define a Library
     4. Alter   - Alter a Library

If List option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria

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

Figure 4-8 Specify a library name to SMS for a non-system managed library

Note: In the Library Name field, enter the SMS friendly name for your tape library. This name will relate your tape library to your SMS tape storage group, which you define later on. There is a minor restriction when you name the library: The first character of a library name must not be `V`, or one of the DFSMSrmm-defined locations (LOCAL, REMOTE, DISTANT).

In the tape library define panel (see Figure 4-9) you have to specify a **Library ID**, the **Entry Default Use Attribute** and **Eject Defaults**. Press Enter to perform verification. Click **END** to save and exit the dialog.

Archived

Description is a 120-byte field that allows you to enter a description of the library definition for use by the installation. There are no restrictions on its content. The use of the description field is optional.

.

```
                              TAPE LIBRARY DEFINE

SCDS Name  . : SYS1.SMS.SCDS
Library Name : NOSMSLIB

To Define Library, Specify:
  Description ==>
              ==>
  Library ID . . . . . . . . . . . FFFFF     (00001 to FFFFF)
  Console Name . . . . . . . . . .
  Entry Default Data Class . . . .
  Entry Default Use Attribute  . . P         (P=PRIVATE or S=SCRATCH)
  Eject Default  . . . . . . . . . p         (P=PURGE or K=KEEP)

  Media Type:          Scratch Threshold
    Media1  . . . . 0              Media3  . . . . 0         (0 to 999999)
    Media2  . . . . 0              Media4  . . . . 0         (0 to 999999)

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

Figure 4-9 Define a new SMS tape library 1 of 2

Note: You can use any library ID not used in your installation.

Check your initial online status settings if you have more than one system sharing this SMS environment. To do so, scroll down the current panel that is displayed and set the **Initial Online Status** for all displayed systems to YES and press Enter.

```
                              TAPE LIBRARY DEFINE                Page 2 of 2
Command ===>

SCDS Name  . : SYS1.SMS.SCDS
Library Name : NORBERTL

  Initial Online Status (Yes, No, or Blank):
    SC63     ===> YES  SC64     ===> YES  SC65     ===> YES


Warning:
  When you connect a tape library to a system group rather than a system,
  you lose the ability to vary that library online or offline to the
  individual systems in the system group.  It is strongly recommended that
  the tape library be connected to individual systems only.
Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 4-10 Define a new SMS tape library 2 of 2

Restart your OAM dynamically by using the MVS command shown in Figure 4-11.

```
MODIFY OAM,RESTART
```

Figure 4-11 Restart OAM to activate the new tape library

Where:

OAM Is the name of the OAM started task

You receive the message shown in Example 4-1 after successfully restarting OAM.

Example 4-1 IGD008I SMS activation message

```
CBR3006I Library NOSMSLIB with Library ID 99999 unknown in I/O
configuration.
CBR3002E Library NOSMSLIB no longer usable.
CBR0097I OAM restart completed.
```

You can use the MVS DISPLAY SMS command as shown in Figure 4-12 to see the newly defined library.

```
D SMS,LIB(ALL),DETAIL
```

Figure 4-12 Use MVS display SMS command

Example 4-2 shows you the result of the MVS display SMS command.

Example 4-2 Output of a DISPLAY SMS command

```
CBR1110I OAM library status: 389
TAPE      LIB  DEVICE    TOT  ONL  AVL  TOTAL  EMPTY SCRTCH  ON OP
LIBRARY   TYP  TYPE      DRV  DRV  DRV  SLOTS  SLOTS   VOLS
LIB1      AL    349-4L1    4    0    0   1445     32      6  Y  Y
NOSMSLIB  UNK              0    0    0      0      0      0  N  N
```

Note: Deleting a tape library using ISMF panels has no effect on the TCDB. Instead, the library definition is removed only from the specified SCDS. To delete a tape library from the TCDB, use the IDCAMS DELETE command.

Defining a tape storage group

To use SMS storage groups, first define tape storage groups using ISMF. To defined tape storage groups you need to have at least one system managed tape library defined to ISMF. See “Defining a tape storage group” on page 153.

Archived

The storage group type TAPE is provided to classify tape cartridges in DFSMS. A tape storage group consists of tape libraries and the tape cartridges associated with them. A tape storage group can contain one to eight tape libraries specified by their library name, and one tape library can contain more than one tape storage group.

Note: You can skip this section if you do not make storage group assignments.

To define a Storage Group following steps are required:

1. Choose Option **6** (**Storage Group**) on the ISMF PRIMARY OPTION MENU to display the STORAGE GROUP APPLICATION SELECTION menu, as shown in Figure 4-6.
2. On that panel, specify the **Storage Group Name** and a **Storage Group Type.** The Storage Group Type is TAPE in our case.
3. Choose Option **2 Define** to display the TAPE STORAGE GROUP DEFINE panel, as shown in Figure 4-13.

.

```
                    STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . 'SYS1.SMS.SCDS'
                          (1 to 44 character data set name or 'Active' )
  Storage Group Name    LOAN              (For Storage Group List, fully or
                                          partially specified or * for all)
  Storage Group Type    TAPE              (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                          OBJECT, OBJECT BACKUP, or TAPE)
Select one of the following options  :
  2  1. List    - Generate a list of Storage Groups
     2. Define  - Define a Storage Group
     3. Alter   - Alter a Storage Group
     4. Volume  - Display, Define, Alter or Delete Volume Information

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

Figure 4-13 Specify a storage group name to SMS

4. Figure 4-14 shows the Tape Storage Group Define panel. The SCDS Name and Storage Group Name fields are output fields containing the SCDS and storage group names, which you specified in the Storage Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the tape storage group. ISMF primes the field with blanks, which are the default.
   The Library Names field is used to specify the tape libraries that own the

volumes within this storage group. One to eight library names can be associated with a tape storage group. At least one library name must be specified when defining a tape storage group. The library name in the tape storage group definition must also be defined in the same SCDS.

5. Press Enter to perform verification.
6. Issue the **END** command to save the newly defined tape storage group and return to the Storage Group Application Selection panel.

```
                          TAPE STORAGE GROUP DEFINE
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : LOAN

To DEFINE Storage Group, Specify:

  Description ==> NON-SMS ON LOAN SCRATCH POOL
              ==>

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

  DEFINE    SMS Storage Group Status . . N  (Y or N)

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

Figure 4-14 Define a new SMS storage group

Repeat step 1 to step 4 for each storage group you have to define.

Note: For each different DFSMSrmm volume pool, you will select by using SMS ACS routines that you have to define a SMS storage group.

Check your SMS storage group status settings if you have more than one system sharing this SMS environment. To do so specify **Y** in the DEFINE SMS Storage Group Status field in the panel shown in Figure 4-14 before you press Enter. The settings for each system should match the settings of your SMS tape libraries. Figure 4-15 shows you the SMS storage group status define panel.

Archived

```
                    SMS STORAGE GROUP STATUS DEFINE          Page 1 of 2
Command ===>

SCDS Name  . . . . : SYS1.SMS.SCDS
Storage Group Name : NORBERTL
Storage Group Type : TAPE
To DEFINE Storage Group System/                    ( Possible SMS SG Status
Sys Group Status, Specify:                          for each:
                                                    - Pool SG Type
 System/Sys      SMS SG   System/Sys      SMS SG       NOTCON, ENABLE, DISALL
 Group Name      Status   Group Name      Status       DISNEW, QUIALL, QUINEW
 ----------      ------   ----------      ------    - Tape SG Type
 SC63       ===> ENABLE   SC64       ===> ENABLE       NOTCON, ENABLE,
 SC65       ===> ENABLE              ===>              DISALL, DISNEW
            ===>                     ===>           - Copy Pool Backup SG Type
            ===>                     ===>              NOTCON, ENABLE )
            ===>                     ===>           * SYS GROUP = sysplex
            ===>                     ===>             minus Systems in the
            ===>                     ===>             Sysplex explicitly
            ===>                     ===>             defined in the SCDS
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 4-15 SMS STORAGE GROUP STATUS DEFINE

If you have successfully defined all your storage groups, you can display them by selecting Option **1 Generate a List Storage Group** in the storage group selection menu (see Figure 4-13). Specify an asterisk (*) in the storage group name field to show all defined storage groups in your SMS environment.

```
                              STORAGE GROUP LIST
Command ===>                                                Scroll ===> HALF
                                                         Entries 1-6  of 27
                                                         Data Columns 3-6 of 43
CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE       STORGRP  SG                VIO      VIO   AUTO
     OPERATOR   NAME     TYPE              MAXSIZE  UNIT  MIGRATE
    ---(1)----  --(2)--- -------(3)------  --(4)--  (5)-  --(6)---
                DB8B     POOL              -------  ----  NO
                DB8BLOG1 POOL              -------  ----  NO
                FKSMF    POOL              -------  ----  NO
                LOAN     TAPE              -------  ----  --------
                OPENMVS  POOL              -------  ----  NO
                PL3490   TAPE              -------  ----  --------
```

Figure 4-16 List all defined SMS storage groups

You activate the new SMS configuration dynamically using the MVS SETSMS command as shown in Figure 4-11.

```
SETSMS SCDS(SYS1.SMS.SCDS)
```

Figure 4-17 Restart SMS to activate the new tape library

Where:

SCDS Is your current used SMS source data set

You get the IDG008I message shown in Example 4-3 after successful activation.

Example 4-3 IGD008I SMS activation message

```
RESPONSE=SC65      IGD008I NEW CONFIGURATION ACTIVATED FROM SCDS
RESPONSE=SYS1.SMS.SCD
```

Define management classes

As no management class attributes are available for tape cartridges, defining management classes is optional. You have to define policies for management classes by defining vital record specifications. Use the RMM subcommand ADDVRS with the DSNAME operand, or the DFSMSrmm Add Data Set VRS panel in the DFSMSrmm ISPF dialog to define vital record specifications. Use the data set name masks that match the management class names you have defined.

Note: You can skip this section if you do not make management class assignments.

In our examples we will define a management class to move the special meaning expiration date decision from the EDGUX100 user exit to SMS ACS. We will define a management classes of M99000 (catalog control). You can change the first qualifier from “M” to “D” if you migrate from EDGUX100 to SMS ACS, and you have previously defined all the vital record specifications to match the special meaning expiration dates starting with a “D”.

Note: There are no differences in the VRS definitions to match to a management class, or to match a VRS management value.

To define a management class, the following steps are required:

1. Choose Option **3** (**Management Class**) on the ISMF PRIMARY OPTION MENU panel as shown in Figure 4-6 to display the MANAGEMENT CLASS APPLICATION SELECTION panel.
2. To define a management class, you must specify a *CDS Name* and a *Management Class Name* on the panel shown in Figure 4-18. The CDS Name

must be the name of an SCDS. ISMF primes the field with the last used name. The default is the quoted word 'ACTIVE', which represents the currently active configuration, but you cannot define or alter management classes to the 'ACTIVE' configuration. In the Management Class Name field, you must specify the name of the management class that you are defining.

```
                    MANAGEMENT CLASS APPLICATION SELECTION        Page 1 of 2
Command ===>

To perform Management Class Operations, Specify:
  CDS Name  . . . . . . . . . . 'SYS1.SMS.SCDS'
                                (1 to 44 character data set name or 'Active' )
  Management Class Name . . . M99000    (For Management Class List, fully or
                                        partially specified or * for all)

Select one of the following options  :
  3  1. List    - Generate a list of Management Classes
     2. Display - Display a Management Class
     3. Define  - Define a Management Class
     4. Alter   - Alter a Management Class

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria


Use ENTER to Perform Selection; Use DOWN Command to View next Selection Panel;
```

Figure 4-18 ISMF management class selection

3. Choose Option **3** (**DEFINE**) on the panel to display the MANAGEMENT CLASS DEFINE panel shown in Figure 4-19.

```
                          MANAGEMENT CLASS DEFINE              Page 1 of 5
Command ===>

SCDS Name . . . . . . . : SYS1.SMS.SCDS
Management Class Name : D99000

To DEFINE Management Class, Specify:

  Description ==> Special meaning expdt 99000 whilecatalog
              ==> for DFSMSrmm only

  Expiration Attributes
    Expire after Days Non-usage  . . NOLIMIT     (1 to 9999 or NOLIMIT)
    Expire after Date/Days . . . . . NOLIMIT     (0 to 9999, yyyy/mm/dd or
                                                 NOLIMIT)

  Retention Limit  . . . . . . . . . NOLIMIT     (0 to 9999 or NOLIMIT)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
```

Figure 4-19 ISMF Panel: Management Class Define

Archived

You can use the optional description field to describe the management class. All other fields are not used for a managment class defined for DFSMSrmm use.

Tip: You do not have to define any parameters in the ISMF definition panels, because the management class attributes are not used by DFSMSrmm.

You can restart your SMS system dynamically by using the MVS SETSMS command as shown in Figure 4-20.

```
SETSMS SCDS(SYS1.SMS.SCDS)
```

Figure 4-20 Restart SMS to activate the new tape library

Where:
SCDS Is your current used SMS source data set

You get the IDG008I message shown in Example 4-4 after successfully activation.

Example 4-4 IGD008I SMS activation message

```
RESPONSE=SC65      IGD008I NEW CONFIGURATION ACTIVATED FROM SCDS
RESPONSE=SYS1.SMS.SCDS
```

Tailoring your SMS management class routines

Update your ACS routine to use an appropriate filter list and logic to assign management classes to data sets on tape. When a volume is opened, DFSMSrmm records the management class name you assign to new tape data sets using your ACS routine.

Figure 3-2 on page 63 shows a management class ACS routine that maps some of the special meaning expiration dates, for example 99000, to a management class name. If there is no match, a managment class of CATALOG will be set.

Example 4-5 Sample ACS management class routine

```
PROC 1 &MGMTCLAS
/*******************************************************/
/* DEFINE FILTER FOR TAPE UNITS                        */
/*******************************************************/
FILTLIST TAPEUNITS INCLUDE(TAPE*,T3420*,T3480*,T3490*,
                           '3420','3480','3490',
                           T3590*,'3590','3592')
/*******************************************************/
/* DATA SET NAME FILTLIST DEFINITIONS                  */
/*******************************************************/
```

Archived

```
FILTLIST MCDB20   INCLUDE(DB2P.DSNDB%.*.%0*.**,
                         DB2T.DSNDB%.*.T0*.**,
                         DB2D.DSNDB%.*.T0*.**)

FILTLIST MCDB21   INCLUDE(DB2P.DSNDB%.*.M1*.**,
                         DB2D.DSNDB%.*.T1*.**,
                         DB2T.DSNDB%.*.T1*.**)

FILTLIST MCDB22   INCLUDE(DB2D.DSNDB%.*.T2*.**,
                         DB2T.DSNDB%.*.T2*.**)
  /*****************************************************/
  /* Support special meaning expiration dates          */
  /*****************************************************/
  IF &UNIT = &TAPEUNITS THEN DO
     WRITE 'MC ACS GETS CONTROL TO SET EXPDT MC'

     SELECT
        WHEN (&EXPDT = '1998001') SET &MGMTCLAS = 'M98001'
        WHEN (&EXPDT = '1999002') SET &MGMTCLAS = 'M98002'
        WHEN (&EXPDT = '1999003') SET &MGMTCLAS = 'M98003'

        WHEN (&EXPDT = '1990000') SET &MGMTCLAS = 'M99000'
        WHEN (&EXPDT = '1999000') SET &MGMTCLAS = 'M99000'

        WHEN (&EXPDT = '1999001') SET &MGMTCLAS = 'M99001'
        WHEN (&EXPDT = '1999002') SET &MGMTCLAS = 'M99002'
        WHEN (&EXPDT = '1999003') SET &MGMTCLAS = 'M99003'
        WHEN (&EXPDT = '1999004') SET &MGMTCLAS = 'M99004'

        OTHERWISE SET &MGMTCLAS = 'CATALOG'

     END
  END
  /*****************************************************/
  /* SELECT VALID MANAGEMENT CLASS                     */
  /*****************************************************/

  SELECT
     /*****************************************************/
     /* Accept special meaning expdt                      */
     /*****************************************************/
     WHEN (&MGMTCLAS = '' && &DSN = &MCDB20) SET &MGMTCLAS = 'MCDB20'

     WHEN (&DSN = &MCDB21) SET &MGMTCLAS = 'MCDB21'

     /*****************************************************/
     /* Accept any expdt                                  */
     /*****************************************************/
     WHEN (&EXPDT = '' && &DSN = &MCDB20) SET &MGMTCLAS = 'MCDB20'
```

```
    END
END
```

Important: If you allow the use of special meaning expiration dates in your computing center, you should specify the entire range of these dates in order to not lose data.

This example can be used for both system managed and non-system managed tape volumes. If you will use the example for non-system managed tape volumes only, you have to include the code shown in Figure 4-6.

Example 4-6 Sample ACS management class routine for tape only

```
IF &ACSENVIR = 'RMMVRS' THEN
      SELECT
         WHEN (&EXPDT = '1998001') SET &MGMTCLAS = 'M99001'
         WHEN (&EXPDT = '1999002') SET &MGMTCLAS = 'M99002'
   ....
END
```

Tailoring your SMS storage group routines

To assign a storage group, you must have the SMS subsystem active, and have a valid SMS configuration.

You use the ACS routines to process the special calls that DFSMSrmm makes to the SMS subsystem for ACS processing. DFSMSrmm requests that the management class and storage group routines are run. The environment variable is set to RMMPOOL so that you can differentiate allocation requests for system-managed data sets from requests by DFSMSrmm for a storage group name. When DFSMSrmm calls the ACS routines with the &ACSENVIR variable, set to either RMMPOOL or RMMVRS; the storage class (&STORCLAS variable) is set to the word USERMM.

Attention: You must use the EDGUX100 exit to request DFSMSrmm to disable the tape drive cartridge loader to prevent specific scratch pool requests from emptying the loaders through volume rejection. This function is not available using SMS ACS routines to select scratch pools.

Example 4-7 shows a very simple example to use a specific scratch pool if the high-level qualifier is MHLRES1 or MHLRES4 for the new tape data set. Both volume pools must be defined in DFSMSrmm.

To make your decisions, you can use all read only variables described in 4.1.4, “SMS read-only variables” on page 143.

Example 4-7 ACS storage group routine for non-system managed tape

```
IF &ACSENVIR = 'RMMVRS' THEN EXIT

IF &ACSENVIR = 'RMMPOOL' THEN DO
   WRITE 'SG ACS GETS CONTROL &ACSENVIR=' &ACSENVIR
   SELECT
      WHEN (&HLQ = 'MHLRES1') SET &STORGRP = 'PL3490'
      WHEN (&HLQ = 'MHLRES4') SET &STORGRP = 'LOAN'
   END
   EXIT
END
```

Tip: The SMS ACS management class routine is called first so you can use this routine to make much more powerful decisions in this routine, and to use the result in the SMS ACS storage group routine.

Tailoring your EDGRMMxx PARMLIB member

Specify SMSACS(YES) in the EDGRMMxx PARMLIB member as shown in Example 4-8 to control whether DFSMSrmm calls SMS ACS processing to enable the use of storage group and management class values with DFSMSrmm.

Add new VLPOOL commands or update your previous defined VLPOOL commands in the EDGRMMxx PARMLIB member (see Example 4-8) to specify the NAME operand. If the NAME is a valid storage group name, it is used as the default STORGRP value for all volumes added into this pool range. You can choose another storage group name when adding a volume by using the STORGRP operand on the ADDVOLUME command.

The same NAME value can be used on multiple VLPOOL commands enabling you to group multiple prefix ranges into a single logical pool without the need to specify STORGRP on ADDVOLUME.

By naming a specific storage group on ADDVOLUME, you can select at the individual volume level to which pool it belongs.

Example 4-8 EDGRMMxx PARMLIB member to support SMS ACS processing

```
OPTION    OPMODE(R)                 /* Record-Only Mode          */  -
          .....
          SMSACS(YES)               /* enable MV ACS processing */   -
          .....
```

```
VLPOOL    PREFIX(F*)                                                    -
          TYPE(R)                                                       -
          RACF(Y)                                                       -
          NAME(PL3490)                                                  -
          EXPDTCHECK(Y)                                                 -
          DESCRIPTION('FOREIGN CARTRIDGES')
VLPOOL    PREFIX(L*)                                                    -
          TYPE(S)                                                       -
          RACF(Y)                                                       -
          NAME(LOAN)                                                    -
          EXPDTCHECK(Y)                                                 -
          DESCRIPTION('TAPES FOR LOAN LOC PROCESSING')
```

Note: For manual tape libraries, DFSMSrmm uses the assigned storage group as the pool name to validate that a mounted scratch volume is from the requested pool.

To restart DFSMSrmm, use the modify command as shown in Figure 4-21 to implement updates to the data set, and avoid stopping and restarting DFSMSrmm. You can also restart DFSMSrmm using another PARMLIB data set member with parameters to implement changes.

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

Figure 4-21 Restart DFSMSrmm after updating EDGRMMxx

Where:

xx Is the suffix of the EDGRMMxx PARMLIB member

You receive the messages shown in Example 4-9 after DFSMSrmm is successfully restarted.

Example 4-9 DFSMSrmm restart messages

```
EDG1101I DFSMSrmm MODIFY COMMAND ACCEPTED
EDG0204I DFSMSrmm BEING INITIALIZED FROM MEMBER EDGRMM03 IN RMM.PARMLIB
EDG0105I DFSMSrmm SUBSYSTEM INITIALIZATION COMPLETE
```

Adding volumes to the volume pool

You can add new volumes to DFSMSrmm using the TSO RMM ADDVOLUME subcommand, or you can use the TSO RMM CHANGEVOLUME subcommand to update details for volumes already defined to DFSMSrmm. For both functions, you can also use the RMM ISPF dialog.

Use the command as shown in Figure 4-22 to add 10 new scratch volumes.

Archived

```
RMM ADDVOLUME LOAN01 COUNT(10) STATUS(SCRATCH) INIT(Y) POOL(L*)
```

*Figure 4-22 Add 10 new scratch volume to pool L**

Where:

COUNT	Specifies the number of volumes to be added
STATUS	Specifies the volume status
INIT	Specifies whether the volume must be initialized before it can be used
POOL	Specifies a POOL where DFSMSrmm stores the volume in the removable media library

Note: 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 are using previously defined volumes, use the command shown in Figure 4-23 to update some volume details.

```
RMM CHANGEVOLUME LOAN01 POOL(L*) STORAGEGROUP(LOAN)
```

Figure 4-23 Update volumes storage group information

Where:

STORAGEGROUP	Specifies the SMS-defined storage group to which the volume belongs.
POOL	Specifies a POOL where DFSMSrmm stores the volume in the removable media library

Defining your vital record specifications (VRSs)

For all SMS management classes you have defined and set in your SMS ACS routines, you must add a vital record specification (VRS) in DFSMSrmm to define retention and movement policies. See *DFSMSrmm Guide and Reference*, SC26-7404, Chapter 5 “Retaining and Moving Your Volumes” for more information about DFSMSrmm VRS processing. A management class vital record specification must be defined as a data set vital record specification, and the data set name mask must match the management class name defined in SMS.

You can add new vital record specifications to DFSMSrmm using the TSO RMM ADDVRS subcommand, or you can also use the RMM ISPF dialog.

Use the commands as shown in Figure 4-24 to define all new data set vital record specifications used in the sample ACS management class routine.

```
/* define VRSs for special meaning expdt processing 98001 to 98365  */
RMM ADDVRS DATASETNAME('M98001') COUNT(1) LASTREFERENCEDAYS
RMM ADDVRS DATASETNAME('M98002') COUNT(2) LASTREFERENCEDAYS
RMM ADDVRS DATASETNAME('M98003') COUNT(2) LASTREFERENCEDAYS
/* define VRSs for special meaning expdt processing 90000 and 99000 */
RMM ADDVRS DATASETNAME('M99000') CYCLES COUNT(99999) WHILECATALOGED
/* define VRSs for special meaning expdt processing 99001 to 99364  */
RMM ADDVRS DATASETNAME('M99001') COUNT(1) CYCLES
RMM ADDVRS DATASETNAME('M99002') COUNT(2) CYCLES
RMM ADDVRS DATASETNAME('M99003') COUNT(3) CYCLES
RMM ADDVRS DATASETNAME('M99004') COUNT(4) CYCLES
/* define VRSs for all other used management classes                */
RMM ADDVRS DATASETNAME('MCDB20') COUNT(20) BYDAYCYCLES WHILECATALOG -
    UNTILEXPIRED SCRATCHIMMEDIATE NEXTVRS(SAVE)
RMM ADDVRS NAME(SAVE) LOCATION(SAVE) COUNT(5)
RMM ADDVRS DATASETNAME('MCDB21') COUNT(21) CYCLES WHILECATALOG -
    UNTILEXPIRED SCRATCHIMMEDIATE NEXTVRS(EXTRA)
RMM ADDVRS NAME(EXTRA) LOCATION(CURRENT) EXTRADAYS(2)
```

Figure 4-24 Define new data set name vital record specifications

Note: You can use all available ADDVRS operands to specify your vital record specification to match SMS managment classes.

4.1.6 Migration considerations

If you are already using DFSMSrmm, use this new function to migrate your EDGUX100 managment value and volume pool selection functions to the DFSMSrmm SMS ACS support as described previously.

Move pooling decisions out of EDGUX100 user exit

You can implement DFSMSrmm scratch pooling using ACS routines based on storage group names. DFSMSrmm provides support for non-system-managed tape and for system-managed manual tape libraries. This support enables pooling at the individual volume level. You assign a storage group name to each volume by:

- Using DFSMSrmm TSO subcommands
- Using pool information that you define with the DFSMSrmm EDGRMMxx PARMLIB VLPOOL command

You use the ACS routines to process the special calls that DFSMSrmm makes to the SMS subsystem for ACS processing. DFSMSrmm requests that the management class and storage group routines are run.

Move VRS management decisions out of EDGUX100 user exit

You use the ACS routines to process the special calls that DFSMSrmm makes to the SMS subsystem for ACS processing. In replacement of the EDGUX100 user exit VRS management value, DFSMSrmm requests that the management class routine is run.

For each new tape data set, DFSMSrmm processing follows the steps described here depending on the circumstances (see Figure 4-25):

- Pre-ACS processing (see “Pre-ACS support as a way to migrate” on page 167 for details):
  - IGDACSXT exit is called, you can implement this exit to set values to &MSPOLICY, &MSDEST, &MSPOOL, and &MSPARM ACS read-only variables before ACS routines are called. Then your ACS routines can refer to these variables.
  - DFSMSrmm calls EDGUX100 installation exit to enable a VRS management value to be returned. The selected value is returned in the &MSPOLICY variable.
- OPEN processing after a device has been allocated:
  - DFSMSrmm calls SMS ACS processing to proceed the management class ACS routine passing environment information.
  - If a management class is not returned from ACS processing, DFSMSrmm calls the EDGUX100 installation exit to enable a VRS management value to be returned.
  - The specified management class or VRS management value is recorded by DFSMSrmm at the data set level.

Attention: The environment variable &ACSENVIR is set to RMMVRS so that you can differentiate allocation requests for system managed data sets from requests by DFSMSrmm for a management class name.

Archive

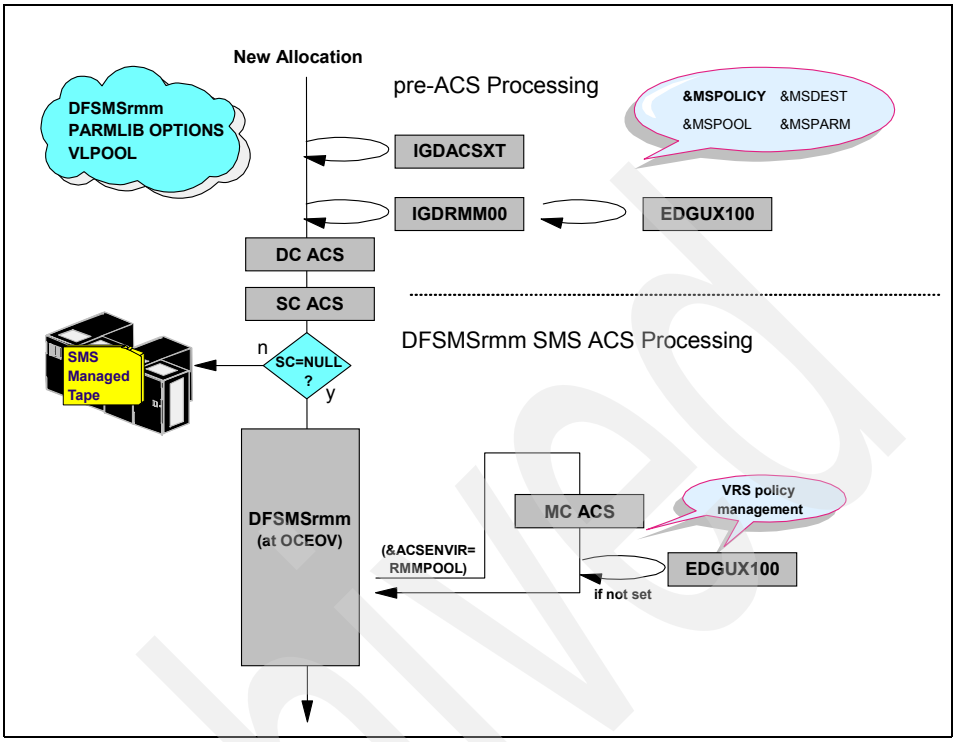


Figure 4-25 DFSMSrmm V2R10 SMS ACS VRS policy management support

Pre-ACS support as a way to migrate

Using the SMS pre-ACS calls to DFSMSrmm, you can use any existing DFSMSrmm and EDGUX100 exit scratch pool and EDGUX100 exit policy assignment decisions as input to your SMS ACS logic. This will enable you to direct new data set allocations to the correct media. You can use SMS ACS to decide if data sets are to be system-managed or non-system-managed. For system-managed tape data sets, you can decide to which library or library type you would like the allocation directed. This can help you with VTS implementation.

You can use the pre-ACS interface to provide additional information like vault destination or pool information to the ACS routines. You can use the EDGUX100 installation exit for pre-ACS processing. When called during pre-ACS processing, values selected by the exit are used as input to the ACS routine.

DFSMSrmm always attempts to pass values for the &MSPOOL and &MSPOLICY ACS read-only variables. If you do not use an EDGUX100 exit, the &MSPOOL variable is set to the DFSMSrmm system-based scratch pool

decision. If you do not use an EDGUX100 exit, there is never a variable set for &MSPOLICY. DFSMSrmm sets a pre-ACS variable only if the variable has not yet been set using the pre-ACS exit. Since DFSMSrmm gets control after the installation exit, any vendor or customer decisions take precedence.

DFSMSrmm can set the following ACS read-only variables:

- **Pool name &MSPOOL**
  Used to specify a tape pool name associated with a data set being allocated. In a system managed tape library, this variable might be used to specify a default storage group where the storage group is equivalent to the tape pool specified in the library.
- **Policy name &MSPOLICY**
  Used to specify a management policy related to a system-managed library

If you only want to use &MSPOLICY and &MSPOOL variables, it is enough to code the EDGUX100 installation exit. EDGUX100 gets control right after IGDACSXT returns the control to the system, and it sets the values only if IGDACSXT did not set them.

In this way, you can use EDGUX100 assignment decisions as input to your SMS ACS logic to enable you to direct new data set allocations to the correct media. The pre-ACS routine passes the address of the ACS installation exit control block ACERO to the EDGUX100 installation exit. The ACERO is mapped by the IGDACERO macro and is the input to the pre-ACS installation exit IGDACSXT. For more information about how you can use the ACERO refer to *z/OS DFSMS Installation Exits*, SC26-7396. You can use any of the values in the ACERO as input to the EDGUX100 installation exit. The sample EDGUX100 exit uses these values:

ACEROJOB	Which is the job name
ACERODSN	Which is the data set name
ACEROEXP	Which is the expiration date. The expiration date is used only if the retention period is not set.
ACERORTP	Which is the retention period

The values are used to perform PL100_CAN_IGNORE, PL100_CAN_VRS, and PL100_CAN_POOL processing. All other functions are performed only when the EDGUX100 is not called by the pre_ACS routine.

If your installation's version of the EDGUX100 installation exit provides values for scratch pooling (PL100_POOL) or vital record specification management value (PL100_VRS), DFSMSrmm updates the ACS input values for MSPOOL and MSPOLICY when they have not already been set by the IGDACSXT pre-ACS installation exit.

Do I use pre-ACS or DFSMSrmm SMS ACS processing

Our intention is to enable you to move function out of the EDGUX100 installation exit. We recommend that you do not use the PREACS option in the EDGRMMxx PARMLIB member.

You can use the existing decisions coded in your EDGUX100 exit to input into your SMS ACS logic. Within system-managed tape, you can decide which library or library type you would like the allocation to be directed. This can help with VTS implementation. Do this by using the pre-ACS processing. Implement or customize the sample EDGUX100 exit (or the EDGCVRSX sample of EDGUX100) to feed pool and VRS management value information into ACS. Base your ACS processing decisions on the MSPOOL and MSPOLICY values that come from DFSMSrmm scratch pool and EDGUX100 exit processing. You can plan to move the EDGUX100 exit decisions into your SMS ACS routines to enable the EDGUX100 exit processing to be ignored or removed.

Using SMS ACS processing, you can consolidate scratch pooling and policy assignment decisions in a single place whether you use system managed tape or not.

4.2 Partitioning a SMS library

Depending on how you want to partition your library, there are different methods.

In any case, you need to use the following resources:

- Scratch categories. DEVSUPxx PARMLIB member
- DFSMSrmm PARMLIB control: REJECT ANYUSE(prefix)
- DFSMSrmm exits: EDGUX100 and EDGUX200
- OAM exit: CBRUXENT

When a cartridge enters the tape library, it may be accepted and assigned a corresponding category by a system, or it may be rejected depending on the partition configuration. Figure 4-26 shows the procedure a system used to decide to accept or reject a volume.

Archive

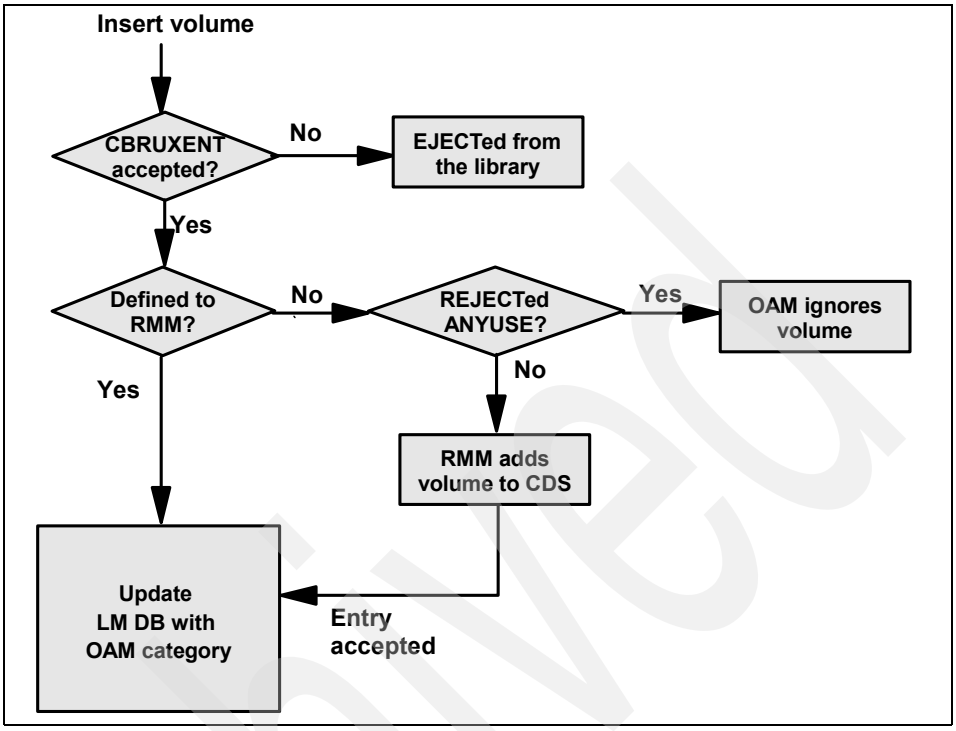


Figure 4-26 Volumes enter a tape library

Note, when you change the CBRUXENT exit, you need to assemble and link edit the module after each maintenance of it.

You base your acceptance of the volumes on your definitions in PARMLIB. The REJECT operand allows you to select the volumes or the volume prefixes allowed in each system. See Example 4-10.

Example 4-10 Using REJECT operand

```
REJECT ANYUSE(A*)     on SYSTEM B
REJECT ANYUSE(B*)     on SYSTEM A
```

With these sentences you are preventing the use of volumes A* on SYSTEM B and volumes B* on SYSTEM A.

In the following sections we document the different library partitioning scenarios that we found. In each of them we explain how partitioning is implemented.

Archived

Partitioning a library between MVS and other systems

You can use DFSMSrmm to manage all your volumes. MVS volumes will be managed automatically and volumes from other systems will be managed manually. Each time you add a volume, specify if this volume will be used from MVS or from another system with the TSO ADDVOLUME subcommand operand as shown in Example 4-11.

Example 4-11 TSO ADDVOLUME subcommand operand USE

```
USE(VM) or USE(MVS)
```

See the command in Example 4-12 to add VM0001 volume to be used only from a VM system.

Example 4-12 Example to use the ADDVOLUME subcommand operand USE

```
RMM ADDVOLUME VM0001 DENSITY(3480) OWNER(YCJRES)-
    STATUS(USER) USE(VM) MEDIANAME(3480)
```

MVS uses only its owns volumes, and DFSMSrmm manages these volumes automatically.

If you prefer, DFSMSrmm has information about the status of the VM volumes; you need to change this information manually using commands.

Partitioning a library with one CDS and one TCDB

The scenario is shown in Figure 4-27. We have one library and two systems sharing the same CDS and the same TCDB.

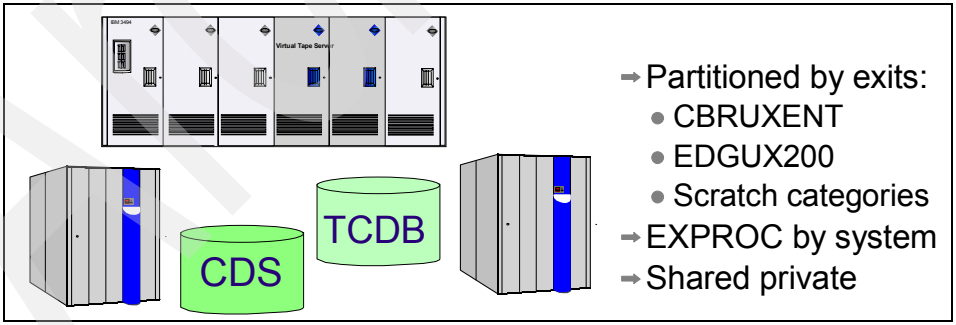


Figure 4-27 Scenario with one CDS and one TCDB

Both the TCDB and CDS are shared between systems; you need to modify CBRUXENT and EDGUX200 exit to partition the library. When a volume that was moved out of the library comes back again because a record for this volume exists in the CDS, then the **REJECT** command in EDGRMMxx will not work.

This kind of partition offers you a convenient way to share private volumes among systems for all the volume entries are in the same TCDB. No additional consideration is introduced by this partition, because both CDS and TCDB contain the information of all SMS volumes.

You need to run EXPROC processing on each system name, and use a customized EDGUX200 exit which prevents volumes returning to scratch on the wrong system. The decisions about which volumes to process should match those coded in the customized CBRUXENT exit.

Partitioning a library with two CDSs and one TCDB

The scenario is shown in Figure 4-28. We have one library, two systems, each system with its own CDS and the same TCDB.

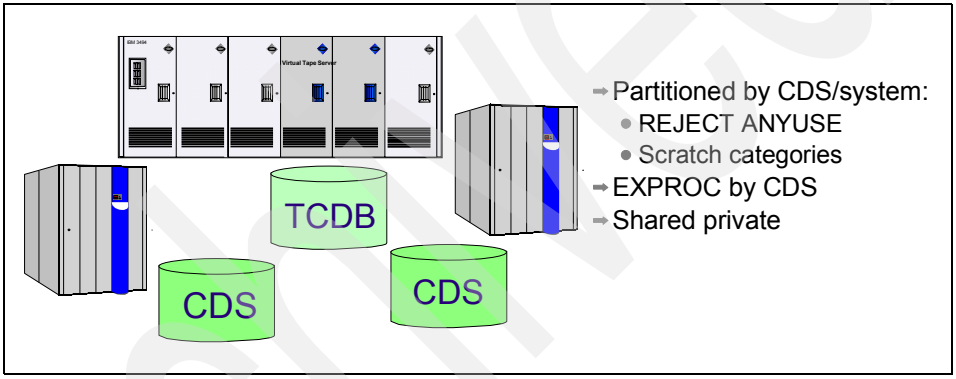


Figure 4-28 Scenario with two CDSs and one TCDB

The tape library is partitioned with CDS. You can use a selective **REJECT** command in EDGRMMxx for DFSMSrmm to partition tape libraries.

You can access all volumes, because the same TCDB is used by both systems. If you want to use private volumes from the other system you need to specify EXPDT=98000 since it is not recorded by the CDS in the system you run the job in.

You need to run inventory management on each system by CDS.

Partitioning a library with two CDSs and two TCDBs

The scenario is shown in Figure 4-29. We have one library and two systems. Each system has its own CDS and TCDB.

Archived

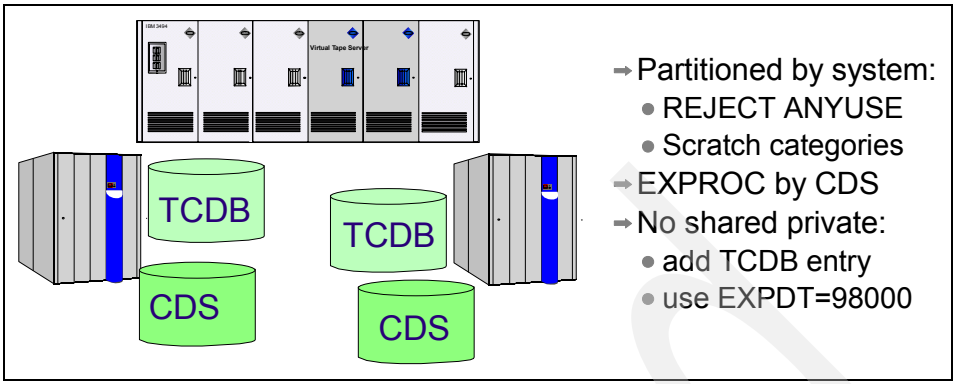


Figure 4-29 Scenario with two CDSs and two TCDBs

Since each system has its own CDS, you can use REJECT ANYUSE(*) to prevent the OAM address space to record information of volumes that you do not plan to use in the specific system.

Private volumes are not shared by systems, you need to define the entry in TCDB when you plan to use a private volume of the other system. You also need to specify EXPDT=98000 when using these volumes, because there is no information about them in CDS. After use, remember to delete the volume entries in TCDB.

You need to run the inventory management task by CDS.

Partitioning a library with one CDS and two TCDBs

The scenario is shown in Figure 4-30. We have one library, two systems, each system with its TCDB and sharing one CDS.

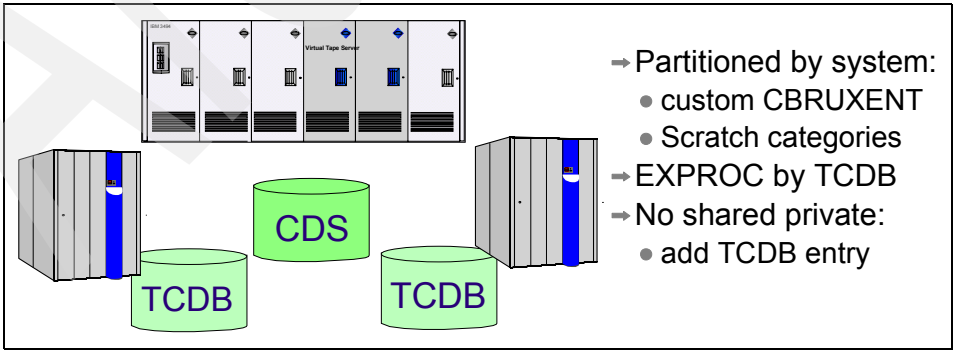


Figure 4-30 Scenario with two TCDBs and one CDS

The CDSs are shared among systems, and the tape library is partitioned with TCDB. You need to modify CBRUXENT to partition a library. If you use the selective **REJECT** command in EDGRMMxx, it may result in a volume being recorded in the wrong TCDB. When a volume is moved out of the library and then comes back again because one record for this volume exists in the CDS, the **REJECT** command in EDGRMMxx will not work.

Private volumes are not shared because each system has its own TCDB. If you want to use private volumes from another system, you must use add volume entry into the TCDB using the ASM utility. You need to delete the entries after work to avoid potential problems.

Although only one CDS is used, inventory management tasks should be run on each system, once for each TCDB, because DFSMSrmm can only return volumes to scratch on a system which has access to the TCDB containing the volume.

4.3 Using DFSMSrmm with virtual tape solutions

Virtual tape solutions are always related with the use of an automated tape library, system managed or non-system managed. Therefore, virtual tape solutions have a different way of working than the traditional tapes. This is why we want to make a separate section for explaining how they work, and how DFSMSrmm provides special support for these solutions and their specific functions.

The DFSMSrmm support for virtual solutions is basically the same for IBM Virtual Tape Server and for other implementations, but there are some specifics for VTS. The interface between DFSMSrmm and OAM is built-in and automated; the interface for other virtual tape solutions must be driven by DFSMSrmm commands.

DFSMSrmm supports virtual tape server import/export in different ways depending on whether or not DFSMSrmm stacked volume support is enabled. When you enable stacked volume support, DFSMSrmm tracks logical volumes and stacked volumes.

To verify that the stacked volume support is enabled, go to the DFSMSrmm Control Record Display panel shown in Figure 4-31.

Archived

```
                        DFSMSrmm Control Record Display
Command ===>
                                                                        More:     +
Type  . . : MASTER      Create date . : 2000/193    Create time . : 17:34:40

Journal utilization . :   0 %    ( 75 % threshold)

Exit status:                              Options:
EDGUX100  . . . : NONE                    Stacked volumes . . . : ENABLED
EDGUX200  . . . : NONE
```

Figure 4-31 DFSMSrmm Control Record Display panel

If it is not enabled, use the EDGUTIL program, as shown in Figure 4-32.

```
//EDGUTIL  JOB <JOB CARD PARAMETERS>
//*
//CREATE   EXEC PGM=EDGUTIL,
//         PARM='UPDATE'
//MASTER    DD  DISP=SHR,DSN=RMM.CONTROL.DSET
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  CONTROL STACKEDVOLUME(YES)
/*
```

Figure 4-32 EDGUTIL example

4.3.1 VTS support for stacked volumes

Prior to OS/390 R10 DFSMSrmm, VTS support was limited to new logical volume type, with the ability to record the container volume for any volume defined to DFSMSrmm. Stacked volumes are not defined nor managed as a stacked volume for DFSMSrmm.

With OS/390 V2R10, DFSMSrmm supports stacked volumes defined to DFSMSrmm. If you do not define stacked volumes, DFSMSrmm defines them automatically at export time if stacked volume support is enabled.

To define stacked volumes, you can use ISPF panels or the RMM TSO subcommand shown in Example 4-13.

Note: You cannot remove stacked volume support after it is enabled. We recommend that you enable stacked volume support whether you have a VTS or not.

Archived

Example 4-13 Using ADDVOLUME with location parameter

```
RMM ADDVOLUME STOOOO TYPE(STACKED) STATUS(MASTER) LOCATION(LIBVTS)
```

When you define a stacked volume, DFSMSrmm checks the TCDB and the library manager. If the volume is not known to the library manager, the volume location is set to SHELF and the destination is set to the VTS library name.

Figure 4-33 shows the VTS support for stacked volumes.

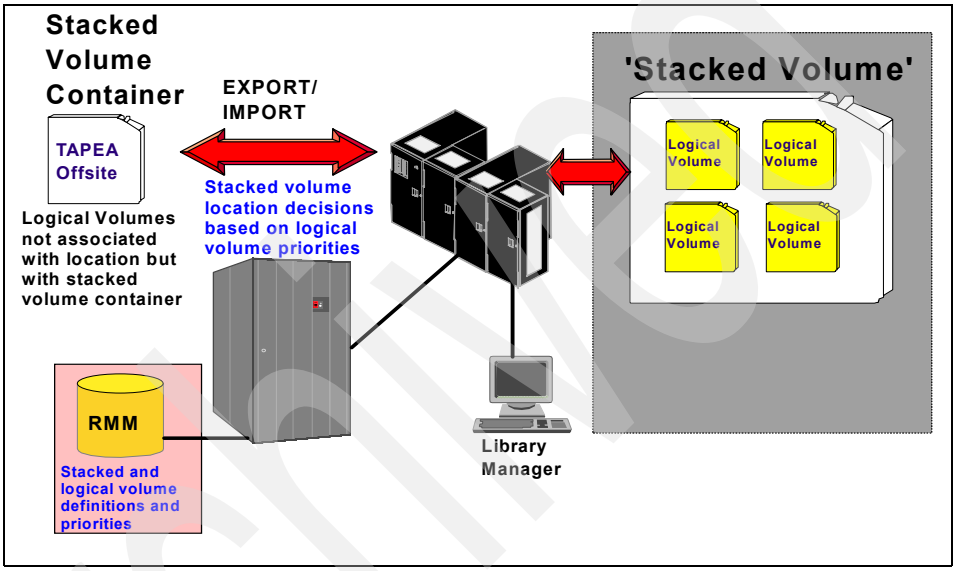


Figure 4-33 VTS support for stacked volumes

At export time, the DFSMSrmm control data set shows that logical volumes are on stacked volumes (field: In container), and stacked volumes are known and managed by DFSMSrmm.

The movement of the stacked volumes is the specified for the logical volumes. DFSMSrmm drives the stacked volume movement based on the required location of the contained volume.

Export processing

The following is an example for exporting a data set IBMUSER.TAPE3490.PRUACS on logical volume MWW000 without previously defining the stacked volume:

1. Define the vital record specification for the data sets you want exported as shown in Example 4-14.

Archived

Example 4-14 TSO RMM ADDDATASET subcommand

```
RMM ADDVRS DSNAME('IBMUSER.TAPE3490.**') DAYS COUNT(3) -
    LOCATION(REMOTE)
```

2. Run inventory management VRSEL processing to determine which logical volumes should be moved and to set the required location for each volume. For our volume, you will get the information in Figure 4-34.

```
Home location  . . . . . : ATLBA999
Required location  . . . . : REMOTE
```

Figure 4-34 VRSEL processing output

3. Example 4-15 shows you how you can create a list of volumes to export.

Example 4-15 Using TSO RMM SEACHVOLUME subcommand

```
RMM SV VOLUME(*) LIMIT(*) REQUIRED(REMOTE) LOCATION(ATLBA999) -
    CLIST('',' REMOTE')
```

4. Start the export process as described in the *DFSMS OAM Planning, Installation, and Storage Administrator Guide for Tape Libraries*, SC35-0427. During this process, DFSMSrmm updates the field “in container” in the logical volume.

   The stacked volume DD0813 is added to DFSMSrmm, with a volume type of STACKED.

5. Run inventory management DSTORE processing to set the destination for the volume.
6. Eject the stacked volume from the VTS from the Library Manager.
7. Confirm that the stacked volume has been moved using command shown in Example 4-16.

Example 4-16 Confirm volume movement

```
RMM CHANGEVOLUME DD0813 CONFIRMOVE
```

Import processing

DFSMSrmm supports the importing of logical volumes that are defined or not to DFSMSrmm. During import processing for a logical volume, DFSMSrmm automatically adds the volume information to the DFSMSrmm control data set or leaves the volume to be processed on another system.

Archived

The following is an example to import a volume:

1. Use the library manager to transfer stacked volume from the “un-assign” category to the “import” category.
2. Create an import list. You can create this list using stacked volumes or logical volumes. To build a list of stacked volumes, use the command in Example 4-17.

Example 4-17 Search for volumes should be imported

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) TYPE(STACKED)-
    DESTINATION(lib_vts) CLIST
```

To build a list of logical volumes to import from a single stacked volume, use the command in Example 4-18.

Example 4-18 List a single volume should be imported

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) OWNER(*) CONTAINER(DD0813)-
    TYPE(LOGICAL) CLIST
```

To build a list of logical volumes to import from multiple stacked volumes, use the command in Example 4-19.

Example 4-19 Search for logical volumes should be imported

```
RMM SEARCHVOLUME VOLUME(*) LIMIT(*) OWNER(*) CONTAINER(*)-
    TYPE(LOGICAL) REQUIRED(lib_vts) CLIST
```

3. Start the import process. If the volume is already known to DFSMSrmm, it is accepted for processing if it is defined as an exported logical volume, otherwise it is rejected.

   During this process, DFSMSrmm removes the stacked volume association from the logical volume.

In Table 4-2 you can see the information that DFSMSrmm tracks in import and export processing for stacked volumes.

Table 4-2 Stacked volume information

Field	Export	Import
Status	NOT EMPTY	EMPTY
Assign Date	Set at start	Set at end
Stacked Count	Increment	Decrement
Use Count	Increment at start	

Field	Export	Import
Usage	Sum of contained volumes	Sum of contained volumes
Security Level	Highest	
Las Read Date		Set
Last Write Date	Set	

You can use DFSMSrmm logical and stacked volumes together with DFSMSrmm commands for ANY virtual tape solution. See 4.3.2, “Stacked volume support for non-VTS” on page 179.

4.3.2 Stacked volume support for non-VTS

You can use stacked volume support for virtual tape solutions different to VTS. The difference is that you must provide to DFSMSrmm the information that OAM provides for VTS during export and import processing.

In this case, you must define to DFSMSrmm stacked and virtual volumes using the RMM commands as shown in Example 4-20.

Example 4-20 Defining non-VTS stacked volumes

```
RMM ADDVOLUME stacked_vol TYPE(STACKED) LOCATION(SHELF)
RMM ADDVOLUME logical_vol TYPE(PHYSICAL/LOGICAL) -
STATUS(MASTER) CONTAINER(stacked_vol)
```

DFSMSrmm is notified by OAM for import and export processing, and can obtain movement completion information from LM; you must do all this using commands.

If you have already defined virtual and stacked volumes, you can add a virtual volume to a container using CV CONTAINER(name). CONTAINER(' ') removes the logical volume from container.

Figure 4-35 shows how DFSMSrmm manages stacked and container, and logical volumes in the export process.

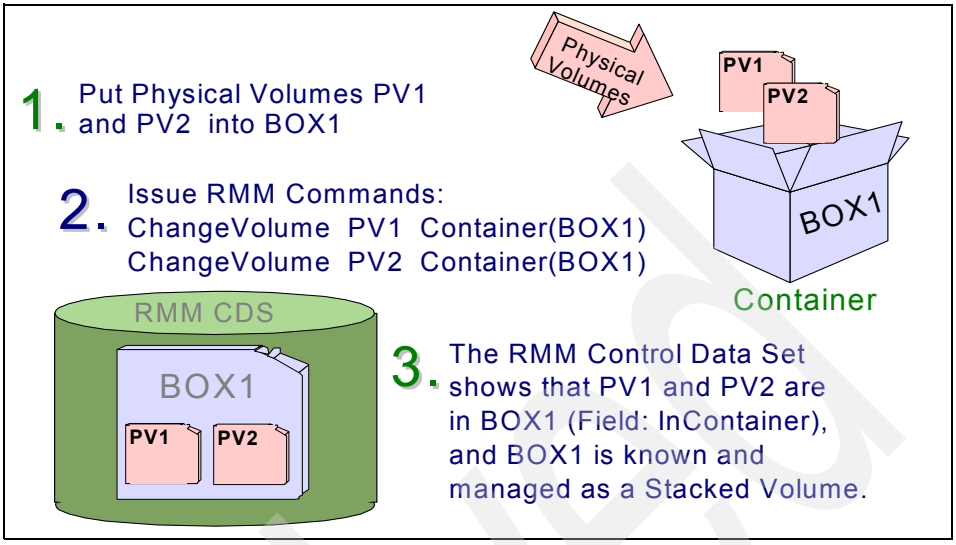


Figure 4-35 Export process for non-VTS

Figure 4-36 shows how DFSMSrmm manages stacked and container, and logical volumes in the import process.

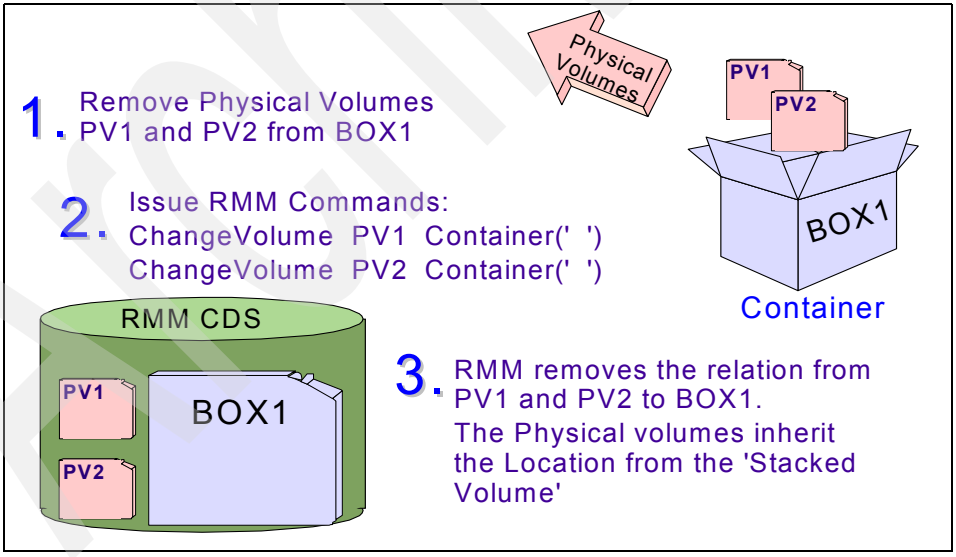


Figure 4-36 Import process for non-VTS

Volumes in a container cannot be moved other than through the container. DFSMSrmm uses the contained volume's required location and priority to

Archived

determine the required location for the container and stacked volume. During VRSEL processing, DFSMSrmm determines where it is necessary to move the stacked volume based on the VRS specified for the logical volume.

Archived

Archived

5

TLMS and DFSMSrmm

To ensure a successful conversion, it is important that you understand some of the differences between the products, and how those differences affect your conversion project.

Objectives of chapter

- Compare DFSMSrmm and TLMS terminology.
- Compare DFSMSrmm and TLMS functions.
- Compare DFSMSrmm and TLMS retention methods.
- Learn which data from TLMS is available for converting, and which is not.

At the end of this chapter you will have a more clear understanding of the correspondence between TLMS and DFSMSrmm management, and also what you can expect from the conversion process.

Archived

© Copyright IBM Corp. 2004

5.1 Terminology

Table 5-1 compares and defines some TLMS and DFSMSrmm terms.

Table 5-1 TLMS and DFSMSrmm terminology comparison

TLMS	DFSMSrmm	Definition
Volume master file (VMF)	Control data set (CDS)	The tape management system database containing information about tape volumes and data sets
Retention master file (RMF™)	- -	The TLMS database containing the management rules of tape data sets. In DFSMSrmm these rules are in the CDS.
Alternate log file	Journal	The sequential data set that keeps track of the changes in the database. It can be used for recovery purposes.
Online recorder	DFSMSrmm subsystem	The address space that controls the real-time processing of the tape management system. It must be active to use the product functions.
Retention type number	Vital record specification (VRS)	The installation-defined policy that manages the life of a tape data set.
Data center (DC)	SHELF location	The location nearest the tape drives.
Cabinet and/or box	Bin number	The individual shelf location in any of the storage locations outside the data center.
PRIVATE volume	MASTER volume	The status of a volume containing active data
Creation date	Assigned date	The date when a PRIVATE or MASTER volume was first added or when a volume changed status either to or from scratch
Keep date	Retention date	The expiration date or calculated by the tape management system
IBM expiration date	Expiration date	The JCL-specified or calculated default expiration date
TRS run	Housekeeping	The batch utility that updates the database and establishes the retention criteria for data sets
Hooks Intercepts	– –	The TLMS user modifications that you must install to interface TLMS with open, close, EOV processing. In DFSMSrmm this interface is integrated within the DFSMS code.

Archive

5.2 Functions

Table 5-2 summarizes the key differences in tape management functions between TLMS and DFSMSrmm.

Table 5-2 TLMS and DFSMSrmm functions comparison

Function	
TLMS	**DFSMSrmm**
Tape security	
► Tape must be scratch for output scratch request ► Data set name on tape must match inventory-specific request ► 44-character data set name checked against VMF on input processing	► Tape must be scratch for output scratch request ► Data set name on tape must match inventory-specific request ► 44-character data set name checked against CDS on input processing ► RACF interface for tape profile management ► Access list for non-RACF system ► Security erase on scratch ► Prevent read of scratch tapes ► Security classification by generic data set name ► Master tape overwrite support controlled by PARMLIB option ► If the DFSMSrmm address space is not started, you cannot use tapes. By using the S DFRMM,OPT=RESET command or the EDGRESET utility, you can permit tapes to be used without DFSMSrmm control.
Retention management	
► Based on RETPD or EXPDT ► By cycle ► By days since last reference ► By days since creation ► By cataloged ► By a combination of cycles and days ► By job name ► Can specify default retention installation wide, and by specific or group data set name ► Retention policies based on full or partial data set name on tape must match inventory-specific request	► Based on RETPD or EXPDT ► By cycle ► By days since last reference ► By days since creation ► By cataloged ► By a combination of cycles and days ► By job name ► Can specify default and maximum retention installation wide. Default retention can also be specified using generic data set mask in VRS. ► Retention policies based on full or generic data set name or VOLSER

Archive

Function	
TLMS	**DFSMSrmm**
Mixed retention management	
	DFSMSrmm allows you to combine any of its retentions together so that a data set must meet all criteria, or the types can be chained in sequence so that a complex and mixed retention can be defined: ► By CYCLE WHILECATALOG DFSMSrmm retains data sets based on cycles or copies of a data set. For a GDG data set VRS, the retention is based on GDG cycles. For non-GDG data sets, each occurrence of a data set is considered to be a cycle. ► By RETPD or EXPDT WHILECATALOG, retains the data set for as long as it is specified in the JCL RETPD parameter. After that period DFSMSrmm retains the data set as long as it is cataloged. ► By days since last reference while cataloged ► By days since creation while cataloged ► By until expired ► By BYDAYSCYCLE (cycles by day)
EXPDT with special meanings	
► 97000-99366	► Dates of 1999365 and 1999366 mean never scratch[1] ► Installation wide exit can be used to support special dates ► Full support of dates beyond twentieth century
Vault or storage location management	
► Supports up to five storage locations [2] ► Define management by data set name – Full or partially qualified – Qualified by job name ► Manages by store bin or slot number and box number ► Unrestricted movement ► Optional qualification by catalog status (while cataloged)	► Supports an infinite number of storage locations and system-managed libraries ► Define management by data set name, VOLSER, or job name – Full or generic (ISMF syntax), GDG or pseudo-GDG ► Manages by store bin number ► Unrestricted movement ► Supports loan location ► Can manage movement based on media name
Security	
► Options table defines authorized users and passwords	► RACF control (through SAF interface) to authorize different functions

Archive

Function	
TLMS	**DFSMSrmm**
Exit for DFSMShsm and Tivoli Storage Manager	
► Supplied	► Supplied ► Full support for DFHSM use of scratch pools
Interfaces	
► Supplied	► Integrated with DFSMS/MVS® ► Can test while TLMS in control
Initialization of new tapes	
► Interface uses WTOR for integrity, updates inventory [3]	► Interface ensures integrity, updates inventory [4]
Label support	
► SL, AL, NL, BLP (BLP requires user exit), NSL	► SL, AL, NL, BLP, SUL, AUL ► For BLP, further processing is determined by the BLP option set in PARMLIB.
Volume pools	
► VOLSER ranges associated with data set name or user support multiple scratch pools based on data set and job name.	► VOLSER ranges associated with system or based on job name and data set name. Installation wide exit EDGUX100 can be used to
Operator messages	
► Scratch mount messages modified with pool name	► Scratch mount messages modified with pool name ► Specific requests modified with rack (slot) number in library ► 3480, 3490, and 3590 message display modified with pool ID or rack (slot) number
Optional sticky labels	
► All, none, non-scratch, out-of-service, out-of-range	► Interface provided through EDGUX100 exit ► Use the DFSMSrmm PARMLIB EDGRMMxx OPTION DISPDDNAME operand and coding the same DD name in a job step that creates or references a tape data set
Identify tapes out of library	
– –	► Loan location support

Archived

Function	
TLMS	**DFSMSrmm**

[1] DFSMSrmm provides permanent retention using 99365/99366 or 1999/365 and 1999/366.
[2] TLMS includes the data center in the count of five.
[3] TLMS provides a modification, TLMSIEHI, to IEHINITT that checks the volume in the TLMS VMF. Also, an operator reply to WTOR is needed.
[4] DFSMSrmm utility EDGINERS includes IEHINITT functions, but DFSMSrmm does not use IEHINITT. DFSMSrmm supports system-managed tape and requires no operator intervention, that is, no WTOR.

5.3 Retention methods

To help you understand the way in which TLMS and DFSMSrmm manage the retention of tape data sets and the movement of volumes between the storage locations, we compare the management rules of the two products.

Table 5-3 presents the TLMS retention methods and the equivalent VRS definitions that you should use in the DFSMSrmm environment.

Table 5-3 TLMS retention methods and DFSMSrmm VRSs equivalences

TLMS	DFSMSrmm VRS
Type 1 - Catalog control	Whilecatalog
Type 2 - Date control	Days or untilexpired
Type 3 - Catalog / date control	Whilecatalog and Days
Type 4 - Cycle control	Cycles
Type 5 - Elapsed days control	Days or extradays
Type 6 - Move immediate	None required
Type 7 - Manual control	Manual move function
Type 8 - Days since last used	Days last reference
Type 9 - Exp. date keyword control	VRS management value and UNTILEXPIRED

The following sections explain the differences between each TLMS retention method and the DFSMSrmm VRS:

Type 1 - Catalog control

The data set is kept until it is uncataloged. TLMS ignores the keep date, and the tape containing this data is scratched when the data set is uncataloged.

In the DFSMSrmm environment the data set is kept while it is cataloged and the expiration date is ignored using RELEASE(EXPIRYDATEIGNORE).

Type 2 - Date control

The data set is retained at the current location until the keep date has passed. The keep date can be determined from the TLMSDTAB table or by specifying a number of days in the count field of the RMF definition. The latter alternative prevents the keep date from being overridden. If the RMF specifies a value, a DAYS retention type is used, and a 1 is added to it to allow for TRS/DFSMSrmm differences; otherwise, UNTILEXPIRED is used so that the volume expiration date is considered.

Optionally, DFSMSrmm ignores the expiration date of the data set, set by the JCL or by installation default, when the RELEASE(EXPIRYDATEIGNORE) operand on the ADDVRS subcommand is specified.

Type 3 - Catalog/Date control

In TLMS the retention method is a combination of Types 1 and 2.

In DFSMSrmm two VRSs are used; the first with DAYS and the second with WHILECATALOG. The DAYS VRS is to be built as a type 2. The VRSs are chained using NEXTVRS.

Type 4 - Cycle control

In TLMS, data sets are retained until the number of versions exceeds the quantity specified.

In DFSMSrmm there is an equivalent retention type specifying CYCLES in the ADDVRS command.

Differences can be found in GDG management processing. TLMS allows the specification of full or qualified data set names, whereas DFSMSrmm allows full generic specification qualified by GDG and non-GDG.

Type 5 - Elapsed days control

In TLMS, data sets are retained at the current location for the number of days specified.

When converted to DFSMSrmm, DAYS is used in the DSNAME VRS, and EXTRADAYS used in NAME VRSs. The number specified in the RMF is increased by 1 to allow for the difference between TLMS TRS and DFSMSrmm VRS.

Type 6 - Move immediate

Tapes are moved from the data center the first time you run TRS after the data set is created.

Archived

In DFSMSrmm this function is available in every VRS definition by specifying the location name desired.

Type 7 - Manual control

In TLMS, a data set is retained until it is either scratched or moved by specific manual instruction.

In DFSMSrmm you can place a volume under manual move control and it remains in a storage location until you manually return it to the data center. For an RMF entry, a VRS with LOCATION(*location*) and DAYS COUNT(99999) is built.

Type 8 - Days since last used

In TLMS, a data set is retained for a specified number of days since it was last used for input or output.

In DFSMSrmm the equivalent is the LASTREFERENCEDAYS operand. The number specified in the RMF is increased by 1 to allow for the difference between TLMS TRS and DFSMSrmm VRS.

Type 9 - Expiration date keyword control

In TLMS, data set retention is controlled by special values coded on the JCL parameter LABEL=EXPDT.

In DFSMSrmm you must use a VRS management value in conjunction with the EDGUX100 exit to manage special meaning expiration dates. At conversion time, EDGCDYNM builds a VRS using UNTILEXPIRED so that any vital record specification management value used for the data set determines retention.

5.4 Data fields

When you display information for a volume or a data set, the data fields differ. In this section we show the differences.

5.4.1 Volume

Table 5-4 compares the TLMS and DFSMSrmm volume data fields.

Table 5-4 TLMS and DFSMSrmm volume data fields

TLMS	DFSMSrmm
Volume chain	Previous and next volume
Keep date	Expiration date

Archived

TLMS	DFSMSrmm
Scratch date	Assign date
Abend indicator	ABEND VRS retention
CPU number	- -
Volume sequence number	Sequence number
- -	Media name
- -	Rack number
- -	Original expiration date
- -	Description
- -	Media type
Recording format (available in TLMS 5.4 but not for 3590 cartridges)	Recording format
- -	Availability
- -	Release actions
- -	Owner
- -	Owner access
Creation date and time	Assigned date and time
- -	Security name and classification
- -	MVS or VM use
- -	Loan location
- -	Volume access list
- -	Volume usage (KB)
- -	Temporary and permanent errors (read and write)
- -	Actions pending
- -	Location type, destination and transit flag
- -	Storage group
- -	Home and required location
- -	Move mode
- -	Bin number and old bin number

Archived

TLMS	DFSMSrmm
- -	Product information
- -	UNTILEXPIRED
- -	EXTRADAYS

5.4.2 Data set

Table 5-5 compares the TLMS and DFSMSrmm data set data fields.

Table 5-5 TLMS and DFSMSrmm data set data fields

TLMS	DFSMSrmm
Abend indicator	ABEND VRS Retention
Create job name and step name	Job name and step name
Last read drive	Included in volume display. Data set display contains creating drive.
Account number	Included in volume display
Data set keep date	Retention date
CPU number	Creating system ID
- -	Owner
- -	Security name and classification

5.4.3 Data unavailable for conversion

The following are new terms in DFSMSrmm that are not available in the TLMS VMF (see 5.1, “Terminology” on page 184 for the field definitions):

- Data set size
- Date data set last read*
- Date data set last written*
- Date volume last read*
- Date volume last written*
- Original expiration date
- Owner
- Rack
- Physical file sequence number
- Security level
- SMS data class name
- SMS storage class name
- SMS storage group name

Archive

- Tape media type
- Tape usage in kilobytes

*In TLMS only the date last used is stored.

5.5 Cross-reference

In this section we present tables that map data from the EDGCNVT input records to TLMS.

During the extract process, the TLMS records are read in input, and some fields are extracted to be converted in the DFSMSrmm database.

Some TLMS fields are not used by the extract program, and some DFSMSrmm fields are not present in the TLMS equivalent record. Also, some fields have a different name in the TLMS database, but have a similar or equivalent meaning in the DFSMSrmm database.

5.5.1 Volume information

Table 5-6 maps data from the EDGCNVT input records to TLMS.

Table 5-6 EDGCLREC volume information

	TLMS	Status
Volume serial number	√	Req
Volume owner [1]		Req
Volume owner user ID		
Volume owner node ID		
Volume security level		
Volume sequence number	√	Req
Volume create date	√	Req
Volume create time	√	
Volume create user ID		
Volume create job name	√	
UCB type (copy of UCBTYP)		

Archive

	TLMS	Status
Tape media type information Tape recording format Not 3480 18 track 36 track 128 track 256 track Tape media type Not 3480 Standard capacity Enhanced capacity High performance cartridge tape Enhanced high performance cartridge tape Tape compaction technique Unknown Not compact IDRC Tape special attributes None 18 track read-only	 √ √ √ 	Cond
Total volume kilobytes used		
JCL volume expiration date	√	Req
Volume expiration date	√	Req
Volume retention date	√	Cond
Count of data sets on volume	√	Req
Temporary read errors	√	
Temporary write errors	√	
Permanent read errors	√	
Permanent write errors	√	
Tape density (copy of JFCBDEN)	√	
BPI indicator 1600 bpi 6250 bpi 3480 3480 compact (IDRC) Undefined	√	Req

Archived

	TLMS	Status
Volume status indicators		Req
Volume is master	√	
Volume pending release		
Vital record - do not release		
User tape (assigned by librarian)		
Tape is on loan	√	
Tape open - not yet closed	√	
Volume is scratch		
Volume recorded by open, close, or EOV		
Additional volume status indicators		Req
Scratch volume claimed by getvol		
Scratch volume has never been initialized		
Scratch volume with init action pending		
Scratch volume waiting to enter ATL		
Abend in process when a data set closed	√	
Abend probably in open, close, or EOV		
Volume label type		Req
Default retention period used		
Label type is NL	√	
Label type is AL	√	
Label type is SL	√	
Tape last written with BLP	√	
SL or AL with user labels	√	
Volume release actions		
Return to scratch pool - default		
Replace tape on release		
Reinitialize		
Degauss/security erase		
Return to owner		
Notify owner		
Volume access information		Req
Owner may read volume		
Owner may update volume		
Owner may alter volume		
Read-only protection		
Update protection		
Can be used on MVS systems	√	
Can be used on VM systems		
No data set recording		
Media name		
Unit address used for creation	√	

Archived

	TLMS	Status
Last used unit address	√	
Loan location	√	
Date volume last read		
Date volume last written		
Assigned from scratch date	√	
Assigned from scratch time	√	
Copy of JFCBLTYP	√	
From JFCTRTCH - IDRC support DSN used 3480 IDRC No compaction	√	Req
Current location of volume	√	Req
Destination name		
Home location name		
Storage group name		
Accounting information	√	
User description		
Authorized user IDs area		
Rack number [2]		Req
Remote store bin number	√	Cond
Remote store bin date	√	Cond
Remote store bin media name		Cond
Label number of first file	√	Req
Name of first data set on volume	√	
First data set's owner's name		
Sequence number of first data set		Cond
First data sets RECFM	√	Cond
First data sets LRECL	√	Cond
First data sets BLKSIZE	√	Cond
First data sets block count	√	Cond

Archived

	TLMS	Status
First data sets kilobytes		Cond
First data sets security level		
DSN1 create date	√	Req
DSN1 create time	√	
DSN1 last read		Cond
DSN1 last written		Cond
DSN1 flag Data set cataloged Abend in progress when data set closed	 √ √	
VRS management value [3]		Cond
SMS Management Class	√	Cond
First file creating step name	√	Cond
First file creating DD name	√	Cond
First file retention date	√	Cond
Name of last data set on volume		Cond
Creating system ID		
Create program name		
Last used program name		
Last used job name		
Last used step name		
Last used DD name		

- Data that can be found in TLMS is marked with a √.
- Data that DFSMSrmm requires is marked with Req.
- Data that DFSMSrmm requires only when the environment exists is marked with Cond.

[1] We recommend that you use the job name or the data set high-level qualifier (HLQ) as the owner.
[2] We recommend that you use the VOLSER as the rack number.
[3] We recommend that you use the other vendor program¢ s “keep date,” if the keep date is a keyword date, and put a single character in front to create a VRS management value.

Archived

5.5.2 Data set name information

Table 5-7 maps data set name information.

Table 5-7 EDGCDREC data set name information

	TLMS	Status
Data set create date	√	Req
Data set create time	√	
Data set name	√	Req
Compaction status Data set used 3480 IDRC No compaction	√	Req
File sequence number on volume		Req
VOLSER holding data set	√	Req
Volume sequence within set of volumes		Req
Unit where data set was written		
Data set RECFM	√	Req
Data set LRECL	√	Req
Data set BLKSIZE	√	Req
Data set block count	√	Req
Data set size in kilobytes		Req
Data set owner name [1]		Req
Logical file sequence number	√	Req
Data set security level		
Data set date last read		Req
Data set date last written		Req
VRS management value [2]		Cond
SMS Management Class	√	Cond
DSN flag Data set cataloged Abend in progress when data set closed	 √ √	
Creating system ID		

Archived

	TLMS	Status
Creating job name	√	Cond
Creating step name	√	Cond
Creating DD name	√	Cond
Retention date	√	Cond
Create program name		
Last used program name		
Last used job name		
Last used step name		
Last used DD name		

- Data that can be found in TLMS is marked with a √.
- Data that DFSMSrmm requires is marked with Req.
- Data that DFSMSrmm requires only when the environment exists is marked with Cond.

[1] EDGCNVT ensures that all data sets on the same volume have the same owner as the VOLSER's owner.
[2] We recommend that you use the other vendor program's "keep date," if the keep date is a keyword date, and put a single character in front to create a VRS management value.

5.5.3 Vital record information

Table 5-8 maps the vital record Information.

Table 5-8 EDGCKREC policy information

	TLMS	Status
Record create date		Req
Record create time		
VRS type		Req
Data set name	√	Cond
Format of data set name	√	Cond
Type of retention Cycles Cycles by days Elapsed days Extra days Days unreferenced While cataloged Until expired	√	Cond

Archived

	TLMS	Status
Volume serial number	√	Cond
Delay days		Req
Total count of number of cycles or days	√	Req
VRS name		Cond
Store keep number	√	Req
VRS owner		Req
Automatic delete date		
VRS description		
Location name	√	Req
Name of NEXT VRS definition		Cond
Job name		Cond
Next VRS options ANDVRS Expiry date ignore Scratch immediate		
Job name	√	

- Data that can be found in TLMS is marked with a √.
- Data that DFSMSrmm requires is marked with Req.
- Data that DFSMSrmm requires only when the environment exists is marked with Cond.

5.5.4 Empty rack and bin information

Table 5-9 shows empty rack and bin information.

Table 5-9 EDGCEREC empty rack and bin information

	TLMS	Status
Shelf status Shelf is rack in library Shelf is bin in storage location		Req
Empty shelf locations name		Req
First of empty shelves start		Req
Count of contiguous shelves		
Media name		

Archived

	TLMS	Status
► Data that DFSMSrmm requires is marked with Req.		

5.5.5 Owner record information

An owner record is required only for actual owners for whom you want to supply owner details, and for those who are to be owners for VRS. Other owner records will be created by CONVERT postprocessing (EDGCVOVL).

Table 5-10 lists the owner record information that DFSMSrmm requires and is not present in the TLMS database.

Table 5-10 EDGCOREC owner information

Description	TLMS	Status
Owner identifier		Req
Owner's surname		
Owner's first name		
Owner's department [1]		Req
Owner's first address line		
Owner's second address line		
Owner's third address line		
Owner's internal telephone number		
Owner's external telephone number		
Owner's user ID		
Owner's node ID		
► Data that DFSMSrmm requires is marked with Req. [1] Not strictly required for conversion. Only the RMM ADDOWNER command enforces the owner's department.		

5.6 Differences

To complete the comparison of TLMS and DFSMSrmm, we must discuss the differences in system structure.

Archived

5.6.1 Structure

TLMS works with two address spaces: CAIRIM and OMS/MONITOR. The CAIRIM address space is the Resource Initialization Manager, which TLMS uses to dynamically initialize and install the TLMS operating system management modules. The OMS/MONITOR address space receives information and controls all tape activity whenever a tape is used. It has a subtask called the Online Recorder, which controls the real-time processing of TLMS, and updates the VMF catalog.

DFSMSrmm has only one address space that manages all tape activity, and the updates to the CDS. DFSMSrmm has a subsystem interface that must be active in order to use tape management functions and ensure tape data set protection. If this interface is defined to MVS in the IEFSSNxx PARMLIB member but is not started, you cannot use tapes in your system. You can also control the inactivation of the subsystem interface using the DFSMSrmm EDGRESET utility through RACF. Thus, you can ensure that no tape activity is allowed unless DFSMSrmm is active.

5.6.2 Open, close, and EOV interface

In a TLMS environment, if you stop the OMS/MONITOR address space, you can use tapes without any knowledge of the tape management system. TLMS requires changes to IBM open, close, and EOV (O/C/EOV) modules to interface tape activity. The changes are accomplished with zaps (called hooks) or dynamic updates to DFSMSdfp modules. You can apply these hooks permanently by using an SMP USERMOD, directly modifying code with the AMASPZAP program, or using the Dynamic Zap Method. The Dynamic Zap Method does not permanently alter the system libraries; the required hooks are dynamically applied to the storage copies of the IBM modules.

In a DFSMSrmm environment, the code that interfaces the open, close, and EOV modules is integrated in DFSMSdfp, so you do not have to make any changes.

5.6.3 Shelf management

TLMS uses *cabinets* and *boxes* to hold tapes offsite. Cabinets are defined with a unique two-character alphanumeric identifier and a slot position for the cartridge. Boxes are defined with a unique four-character alphanumeric identifier and a number indicating the total capacity of the box. You can move complete boxes only, not individual tapes, from one storage location to another.

Archived

In DFSMSrmm, bin numbers define individual shelf locations in offsite storage locations. Bin numbers are six characters and can only be assigned by DFSMSrmm that use a sequential filling method of the bins in the selected location. Once a volume no longer uses a bin number, DFSMSrmm can reuse that number for another volume that needs to be placed in the same location. Thus, you do not have to worry about the physical location of the tape in the offsite storage location, because DFSMSrmm tells you which free bin number you can use for the volumes that you are moving.

5.6.4 Retention

In most cases, TLMS uses a control file, which is the first file on the tape and drives the retention of all other files on the volume. The retention method is applied only to the first data set; your application must write the other data sets according to the retention method applied to the first.

DFSMSrmm applies a VRS definition to every data set on the volume, according to the data set or job name mask defined in the VRS, without considering the file position on the tape. This method should be carefully considered when planning the conversion process, because the VRS definition generated could be different from the management that you want.

DFSMSrmm can return a single volume in a multi-volume set of volumes to scratch; it does not always operate on the complete set of volumes. Consequently, you may see DFSMSrmm return a volume to scratch that is still retained by TLMS. Each data set in a multi-volume set can be retained individually using VRS. If there are data sets in the volume set that have different retention requirements, DFSMSrmm can attempt to release them earlier.

For a volume to be released, all data sets on that volume must no longer be retained by VRS. If you use a control file to cause the complete set to be retained, you may require additional VRS definitions under DFSMSrmm to cause the same volumes to be retained. Alternatively, you can benefit from DFSMSrmm releasing volumes early when the data is not required to be retained.

5.6.5 Deletion policies

When you return a volume to scratch (manually, or by the inventory management process) in the TLMS database, all data set records, except file one, are deleted immediately, so you cannot recover the other data set.

In the DFSMSrmm CDS, the data set records are not deleted. They are deleted only when the volume is reused. Therefore, you can reclaim volumes from scratch and still have all data set details.

Archived

6

Data extraction

In this chapter we detail the information that you need to collect from your TLMS environment and the running of the programs that extract the required data for conversion.

Objectives of chapter

- Understand your TLMS environment.
- Understand the necessary information for the conversion programs.
- Learn how to extract information from your current TLMS environment.
- Consider all the possible interfaces between the tape management system and other products.

At the end of this chapter you will have all the necessary data extracted from TLMS and which is ready to be loaded into the conversion programs.

Archived

© Copyright IBM Corp. 2004

6.1 Samples and programs

In addition to this book and the DFSMSrmm publications listed in “Related publications” on page 697, sample Jobs, REXX execs, and programs are available to help you convert to DFSMSrmm in SYS1.SAMPLIB.

The sample JCL is set up to work in a typical installation. Some of the statements in the sample JCL are commented out so that you can choose to use them or not, as you prefer. Procedures, which are normally part of other IBM products, are used to assemble and compile the sample code. You do not have to assemble, link, and compile code before you can use it. If these procedures are not available to you, you can edit the JCL to uncomment the statements that execute the programs from a STEPLIB. If you choose to link-edit the programs yourself, you can use the standard link-edit options. None of the programs are re-entrant.

Note: Please refer to your SAMPLIB member EDGCMM01 for the most recently updated information about samples and programs.

6.1.1 Sample data extraction jobs provided with DFSMSrmm

Table 6-1 lists the sample jobs provided with DFSMSrmm that are used to extract data as part of the conversion process.

Table 6-1 DFSMSrmm provided sample jobs

Name	Location	Function
EDGJCNVT	SAMPLIB	JCL to execute EDGCNVT and EDGCEBIN
EDGJDYNM	SAMPLIB	JCL to execute EDGCDYNM
EDGJACS	SAMPLIB	JCL to execute EDGRACS

6.1.2 Sample extraction REXX execs provided with DFSMSrmm

Table 6-2 lists the conversion REXX exec provided with DFSMSrmm to build the CDS and create the control record.

Table 6-2 DFSMSrmm-supplied conversion REXX execs

Name	Location	Function
EDGRACS	SEDGEX1	REXX source for UXTABLE to SMS ACS translation

6.1.3 Conversion programs and exits provided with DFSMSrmm

Table 6-3 lists the conversion programs and exits provided with DFSMSrmm to build the CDS and create the control record.

Table 6-3 DFSMSrmm-supplied conversion programs and exits

Name	Location	Function
EDGCNXIT	SAMPLIB	Source for EDGCNVT user exit EDGCNXIT
EDGCNVT	LINKLIB	Program to convert data to DFSMSrmm format
EDGCDYNM	SAMPLIB	Source for TLMS extract program
EDGCEBIN	SAMPLIB	Source for EDGCEBIN program to create empty BIN records
EDGCVOVL	LINKLIB	Program to complete owner records
EDGUTIL	LINKLIB	Program to initialize and update the CDS control record and validate and correct the contents of the CDS

6.1.4 Macros provided with DFSMSrmm

Table 6-4 lists the macros provided with DFSMSrmm to build the CDS and create the control record.

Table 6-4 DFSMSrmm-supplied macros

Name	Location	Function
EDGCDREC	MODGEN	Macro that maps the input data set record to EDGCNVT
EDGCEREC	MODGEN	Macro that maps the input empty shelf record to EDGCNVT
EDGCKREC	MODGEN	Macro that maps the input VRS policy record to EDGCNVT
EDGCLREC	MODGEN	Macro that maps the input volume and first data set record to EDGCNVT
EDGCOREC	MODGEN	Macro that maps the input owner record to EDGCNVT

6.1.5 Sample exits described in this book

Sample exits in Table 6-5 are provided with DFSMSrmm. Some of these exits require changes for running DFSMSrmm in parallel to TLMS. We provide the source of the modified exits as additional material with this redbook. For information on how to get these exits, refer to Appendix D, “Additional material” on page 679.

Table 6-5 DFSMSrmm provided sample exits

Sample exit	Function
IGXMSGEX	Programming Interface to EDGMSGEX

Archive

Sample exit	Function
CBRUXENT	Programming Interface to EDGLCSUX
CBRUXEJC	Programming Interface to EDGLCSUX
CBRUXCUA	Programming Interface to EDGLCSUX
CBRUXVNL	Programming Interface to EDGLCSUX

6.2 Reviewing the current environment

See Figure 2-1 on page 51 to review the flow of events that we followed in converting from TLMS to DFSMSrmm. You must collect and then analyze all information about the current tape management environment.

Once you have reviewed the environment and understand the main differences between TLMS and DFSMSrmm, you will better understand what you must change during the conversion process.

6.2.1 Collecting information

The information that you collect about the current environment should include:

- TLMS reports used daily or occasionally

  Every day there should be a TLMS report that the operator uses to move the cartridges between the desired locations and to return tapes to the scratch pools. You can use the DFSMSrmm supplied reports, but if you want to avoid changes to the layout of this report, using REXX or DFSORT you can tailor the DFSMSrmm reports to make them similar to the TLMS reports. *DFSMSrmm Primer,* SG24-5983, provides some useful examples how to create the reports.

- Exits installed

  Collect information about TLMS exits installed in your environment so you can ensure that you can achieve the same function in the DFSMSrmm environment. For example, you must check the OAM exits, CBRUXENT and CBRUXEJC, if you have an SMS managed IBM ATL installed.

- PARMLIB options used

  You must review TLMS PARMLIB options before you start DFSMSrmm because you may have to modify the DFSMSrmm options according to the TLMS environment.

- Interfaces with other products

Archived

You want to collect information about the interfaces that TLMS might have with other products (such as CICS®, ROSCOE, and TSO).

- Housekeeping (batch inventory update) process

  Collect information on the daily inventory management process so that you can synchronize it with the DFSMSrmm housekeeping function.

6.2.2 Analyzing the record layouts

Now you need to analyze the information you have collected to achieve the equivalent or acceptable alternative function in the DFSMSrmm environment.

DFSMSrmm requires certain data to be available to build its CDS. You must analyze TLMS databases and reports to know which types of data are available. There is not a one-to-one correspondence between DFSMSrmm and TLMS data. DFSMSrmm may have data that TLMS does not have, and vice versa. If DFSMSrmm requires data that TLMS does not provide, use the extract programs or the EDGCNVT SYSIN fill statements to provide a value.

After analyzing the data, you must determine which types of data you want to have in the DFSMSrmm CDS in order to be consistent with the current operating environment. Knowing the types of data needed helps determine whether you can use TLMS databases or reports as the sole input to the extract program. Refer to 6.6, “TLMS data extraction” on page 243 for a functional overview of the extract program.

In addition, you must review the current policies regarding retention management, storage location management, and tape pool management. DFSMSrmm manages these processes differently from TLMS.

6.2.3 TLMS PARMLIB options

During the analysis process, you must check the TLMS PARMLIB options that can affect the DFSMSrmm environment. Carefully review all TLMS parameters before proceeding to the next step.

You must modify the DFSMSrmm PARMLIB options to reflect the same defaults as specified in the TLMS options. The most important DFSMSrmm options for the conversion are:

- OPTION statement
  - OPMODE parameter
  - RETPD parameter
  - BLP parameter
  - VRSJOBNAME parameter

  - UNCATALOG parameter
  - VRSEL parameter
  - MASTEROVERWRITE parameter
  - TPRACF parameter

- VLPOOL statement

  Only if you use multiple scratch pool support.

- LOCDEF statement

  Only if you use location names different from the installation defaults.

Use the TLMSOPTS program to automatically produce a listing of TLMS options. Example 6-1 shows the sample output.

Example 6-1 Sample TLMSOPTS output

```
your company name                    INSTALLATION OPTIONS MAINTENANCE PROGRAM      TLMSOPTS    PAGE    2
CA-DYNAM/TLMS 5.4 9506TL54                                                              09 006 1995 11.59.03

                        OPTIONS SETTING LISTING
                       ------------------------------------------------------------------------

                        DATACENTER=DC
                        DATE=MM/DD/YY
                        DBLDRIVE=NO
                        DBLTIME=000000
                        LABELS=NO
                        LOGID=240
                        MODIFY=YES
                        OVERRIDE=NO
                        PAGESIZE=58
                        PROTECT=ALL
                        QSIZE=20
                        RECOVERY=ALTLOG
                        ROUTEAUX=2
                        ROUTEINQ=14
                        ROUTELBL=13
                        SMC=NO
                        SUBTASK=NOAUTO
                        VSNREQD=YES
                        WAIT=20
```

Here are the options to check:

DATACENTER=___ Is the name of the location nearest the tape drives. In DFSMSrmm the equivalent is the SHELF location.

DATE=___ Is the date format used. You can specify the same date format used by TLMS in the DFSMSrmm OPTIONS with the DATEFORM parameter.

DBLDRIVE=___ Indicates whether or not you want to ensure that double-opens of a data set are on the same physical drive. There is no equivalence in DFSMSrmm.

Archived

DBLTIME=___ Tells how much time you want between two opens for output for the same data set. There is no equivalence in DFSMSrmm.

LABELS=___ Indicates whether or not you want to print gummed labels. In DFSMSrmm you can use the EDGUX100 exit.

LOGID=___ Is the SMF record type used by TLMS if the RECOVERY=SMF options in effect. The default is 240. You can use the same record type for DFSMSrmm auditing and security SMF records, but be careful when selecting records for processing during times when both products are running.

MODIFY=___ Indicates whether or not you want to use the OMS/MONITOR subsystem command interface. An equivalent interface is already integrated in DFSMSrmm.

OVERRIDE=___ Indicates whether or not to allow the operator to override the data set protection feature when a non-scratch volume is detected for use as an output data set. In DFSMSrmm the equivalent is the MASTEROVERWRITE parameter of the EDGRMMxx PARMLIB member.

PAGESIZE=___ Sets the maximum number of lines on a printed page. In DFSMSrmm the equivalent is the LINECOUNT parameter of the OPTION control statement.

PROTECT=___ Restricts the "crash-protected" processing of TLMS to selected data sets. In DFSMSrmm you can control this process with the ABEND and OPEN VRS retention policies.

QSIZE=___ Is the size of the CSA storage area that will contain TLMS messages. There is no equivalence in DFSMSrmm.

RECOVERY=___ Indicates whether to create backup records and where to write them. In DFSMSrmm you use journal records and the EDGHSKP or EDGBKUP utilities for backup and restore purposes.

ROUTEAUX=___ Sets a route code to use for the auxiliary message console. There is no equivalence in DFSMSrmm. However, you can set a route code for every DFSMSrmm message by using the MVS MPF facility.

ROUTEINQ=___ Sets a route code to use for the inquiry console. There is no equivalence in DFSMSrmm. However, you can set a route code for every DFSMSrmm message by using the MVS MPF facility.

Archived

ROUTELBL=___ Sets a route code to use for the gummed-labels printer. There is no equivalence in DFSMSrmm.

SMC=___ Indicates whether or not you want to use the system message control option. There is no equivalence in DFSMSrmm.

SUBTASK=___ Indicates whether or not you want to start the inquiry module automatically with the online recorder. In DFSMSrmm, all functions are integrated in the same started task.

VSNREQD=___ Indicate whether or not you want to require that a VOLSER be supplied before the volume is opened for output. There is no equivalence in DFSMSrmm.

WAIT=___ Selects a time interval for the wait time to avoid S522 abends. There is no equivalence in DFSMSrmm.

You must pay attention to the TLMSDTAB table that specifies the default retention days for the data sets not covered by an RMF definition. The default retention is three days. You must modify the DFSMSrmm option RETPD to reflect the same default retention period as the TLMS TLMSDTAB definition.

6.2.4 Inventory management and reports

TLMS provides batch facilities to perform the following functions:

- Batch update and reporting
- Tape retention and expiration
- Storage location management
- Backup, recovery, and verification of databases

Except for batch update and reporting, DFSMSrmm provides similar functions with the EDGHSKP, EDGBKUP, and EDGUTIL utilities. EDGHSKP must be executed with the DFSMSrmm subsystem active. EDGBKUP and EDGUTIL can be executed independently of the subsystem.

The utilities have the following functions:

- EDGHSKP
  - Processes vital records
  - Performs storage location management
  - Performs expiration processing
  - Creates an extract data set
  - Backs up the DFSMSrmm CDS
  - Backs up and clears the DFSMSrmm journal
- EDGBKUP

  - Backs up the DFSMSrmm CDS
  - Backs up the DFSMSrmm journal
  - Recovers and restores DFSMSrmm CDS
  - Reorganizes DFSMSrmm CDS
- EDGUTIL
  - Verifies DFSMSrmm CDS
  - Mending the DFSMSrmm CDS
  - Synchronizes DFSMSrmm CDS to the user catalogs
  - Performs consistency checking between the DFSMSrmm CDS, the TCDB, and the library manager database in a system-managed tape library.

DFSMSrmm does not provide any batch update utility. However, you can execute the DFSMSrmm TSO commands ADD, CHANGE, and DELETE in batch to achieve the same result. The ISPF dialog has the option to generate and group together TSO commands for later batch processing. For more details about the DFSMSrmm utilities, the ISPF dialog, and the TSO commands, refer to the redbook *DFSMSrmm Primer,* SG24-5983, or to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

TLMS provides batch reports. These may be supplemented with the use of the optional CA-EARL product, which allows you to customize your own reports from the DBD.

DFSMSrmm provides EDGRPTD, and EDGAUD reporting programs, and EDGRRPTE reporting REXX. You can execute the DFSMSrmm TSO commands, LIST and SEARCH, in batch to produce a wide variety of reports. You can also use DFSORT's ICETOOL to write your own customized reports.

You also have the option of using the DFSMSrmm report generator to produce required reports.

For more information about the DFSMSrmm reporting programs, REXX, and the use of the DFSMSrmm report generator, refer to the redbook *DFSMSrmm Primer,* SG24-5983, or *DFSMSrmm Reporting*.

The TLMS scratch list report is shifted one day in the DFSMSrmm environment because the DFSMSrmm report lists volumes that are in PENDING RELEASE status. Those volumes are then returned to SCRATCH status during the next housekeeping process, which usually runs next day.

Archive

We suggest that you move to the scratch pool the cartridges that will return to scratch only when they are no longer in PENDING RELEASE status. You can accomplish this in your batch housekeeping jobstream by using a step that lists the volumes in PENDING RELEASE status before you run the EDGHSKP step. These are the volumes that the TLMS scratch report lists as *new scratched volumes* because of the DFSMSrmm PENDING RELEASE status.

Alternatively, you can execute EDGHSKP with the EXPROC parameter a second time to return pending release volumes to scratch. If you specify the RELEASE(SCRATCHIMMEDIATE) operand on the ADDVRS subcommand and the VRSEL(NEW) PARMLIB option, only a single run of inventory management EXPROC processing is required to return volumes to scratch. When the VRSEL(OLD) PARMLIB option is in use, DFSMSrmm ignores any VRS release options that may have been set for a volume.

6.2.5 Exits

DFSMSrmm provides source code for the installation-wide exits installed as part of DFSMSrmm during the SMP/E APPLY process. The source code is supplied for:

IGXMSGEX	DFSMSrmm uses the DFSMSdfp MSGDISP exit to update tape drive displays
CBRUXENT	Object Access Method (OAM) library control system (LCS) cartridge entry exit
CBRUXEJC	OAM cartridge eject exit
CBRUXCUA	OAM cartridge change use attributes exit
CBRUXVNL	OAM volume not in library exit
EDGUX100	DFSMSrmm installation-wide exit; two samples are provided: EDGUX100, and EDGCVRSX in SAMPLIB
EDGUX200	Return a volume to scratch status in an external inventory. Prevent a volume from returning to scratch status. Request that DFSMSrmm ignores data set information recorded for a volume.

Before the SMP/E ACCEPT, the source code is in SMPSTS; after the ACCEPT, the exits are in AEDGSRC1.

If your installation is already using one or more of these exits in the TLMS environment, you must ensure that the correct action is performed during conversion. Chapter 8, “Parallel running and validation” on page 309, and Chapter 9, “Cutover to production” on page 357 describe the procedure to follow.

TLMS uses the CBRUXENT exit to interface the IBM 3494 and IBM 3495 Automated Tape Library Dataservers. You can test the DFSMSrmm version of this exit once you have completed the conversion.

The CBRUXVNL exit is optional. If you use it, however, as with the CBRUXENT exit, you must test it in the DFSMSrmm environment after conversion.

The installation-wide exits that you installed with DFSMSrmm for either IGXMSGEX, CBRUXENT, CBRUXEJC, and CBRUXCUA can replace any existing exits you had previously installed. Decide whether you still require the function your previous exits provided, even if for a short time while introducing DFSMSrmm. If you no longer need the function, proceed with the next implementation step. If you still require the function, consider how to implement it, while still providing the function required by DFSMSrmm.

See the *DFSMSrmm Implementation and Customization Guide*, SC26-7405, for more details on the DFSMSrmm programming interface that you use with EDGTVEXT, IGXMSGEX, and the OAM exits. The supplied source code for these exits shows how to call the programming interface. You can either modify the supplied exit to call a modified version of your old exit, or modify your old exit to call the DFSMSrmm programming interface. Consider how you might provide an option to disable your code without another code modification. In either case, install the changed software by using an SMP/E USERMOD.

Based on the decisions you have taken about VRS management values and scratch pooling, you can update the supplied sample installation exit, EDGUX100, and the dynamic table produced by the conversion programs, to perform the function you require.

We recommend you use the sample EDGUX100 exit that is shipped with DFSMSrmm as SAMPLIB member EDGCVRSX.

For more information about DFSMSrmm exits, refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

6.2.6 Data set overwrite

TLMS supports two volume statuses:

PRIVATE The volume contains valid data.
SCRATCH The volume can be used for scratch output requests.

In an DFSMSrmm environment you have three different volume statuses:

MASTER A private volume that contains data.

SCRATCH The volume can be used for scratch output requests.

USER The volume is assigned to a user, and it can be overwritten, rewritten, and used repeatedly until the user returns it to the scratch pool.

DFSMSrmm can manage the overwrite of a MASTER volume with the PARMLIB option, MASTEROVERWRITE. You can tailor this option to meet your installation's needs. The possible values are:

ADD New data can be created, but no existing data can by destroyed.

LAST Only the last file on a MASTER volume can by overwritten. The data set name must match the existing data set name. This option provides the equivalent function available with TLMS.

MATCH An existing file can be overwritten. The data set name must match the existing data set name. All files which are higher sequence are destroyed.

USER The same as a volume in USER status.

Refer to the *DFSMSrmm Guide and Reference*, SC26-7404, for a detailed explanation of the function.

6.2.7 BLP and NL processing

In a TLMS environment, because NL tapes do not have internal labels and BLP tapes bypass standard labels, you must supply the VOLSER in the JCL or from the operator console.

BLP and NL processing differs in an DFSMSrmm environment. All scratch tapes must be standard label. When DFSMSrmm successfully reads the VOL1 label for a volume mounted as a result of a nonspecific request, it provides the actual VOLSER to OPEN for use in related processing such as cataloging of the data. This replaces the logical VOLSER (Lnnnnn) generated by OPEN in these circumstances.

For BLP, further processing is determined by the BLP option set in PARMLIB. Processing is as follows:

RMM This is the default and allows BLP processing of USER status tapes and BLP input from MASTER status tapes.

NORMM BLP can be used for reading and writing of master and user status tapes, and for output to scratch tapes. BLP reading of scratch tapes is not supported.

When BLP or NL is used with scratch tapes, DFSMSrmm still changes the volume to master status, but sets the initialize release action to ensure that the volume has a valid standard label before it is returned to scratch status.

Some changes are required for how and when DFSMSrmm records information for tapes and data read and written using BLP:

- For BLP output to a nonspecific volume, the file sequence number is not checked. For example, LABEL=(2,BLP) can be used. For non-BLP requests, the restriction of using LABEL=(1,label_type) is still enforced.
- The data set information for files processed with BLP is only updated for output to the first file on a volume. All other types of BLP requests change only the volume information such as the date last read and date last written. For BLP output requests, the data set information is only updated in the DFSMSrmm CDS when the data set is closed.

You can enable the use of NL output to scratch volumes by creating the profile *STGADMIN.EDG.NOLABEL.volser* in the RACF FACILITY class. Once the profile is created, the function is enabled, and only those who have access to the profile can use this function. DFSMSrmm normally supports only standard label scratch tapes for NL tapes, and relabels them at time of use to NL and standard label at return to scratch.

To use NL scratch tapes, use the system tape exit IFG019VM. DFSMSrmm now ships a sample IFG019VM exit as member EDG019VM in SAMPLIB. The sample exit checks for a supported request and issues a WTOR to the operator to obtain the volume serial number. After installing the exit, testing should be done to ensure that the exit supports the installation' needs for NL tape.

6.2.8 DFSMShsm interface

TLMS and DFSMSrmm process the release of DFSMShsm tapes differently. When TLMS is informed through the ARCTVEXT exit that a DFSMShsm volume is no longer required, it uses the extended EDM retention value from PARMLIB to calculate the retention date for the volume. When DFSMSrmm is informed, through the EDGTVEXT general-use programming interface that a DFSMShsm volume is no longer required, the volume is processed according to the following option setting within the DFSMSrmm PARMLIB:

TVEXTPURGE(RELEASE | EXPIRE | NONE)

Specifies how DFSMSrmm processes volumes to be purged by callers of EDGTVEXT or EDGDFHSM:

NONE DFSMSrmm take no action for volumes to be purged.

RELEASE	DFSMSrmm releases volume to be purged according to the release actions set for the volume. You must run expiration processing to return a volume to scratch status.
EXPIRE	Use the EXPIRE option to set the volume expiration date to the current date for volumes to be purged. Use this operand in combination with vital record specifications that use the UNTILEXPIRED retention type. You can then optionally extend retention using the extra days retention type.

6.2.9 Tape initialization

TLMS provides an interface to IEHINITT, the initialization utility. This interface consists of a hook that passes control to a TLMS routine when a tape initialization request is received.

If the VOLSER is not yet present in the DBD, the initialization request is allowed.

If the VOLSER is known, two situations are possible:

- SCRATCH volume: The operator is prompted to allow or not allow the initialization of the volume.
- PRIVATE volume: TLMS sends a message to notify the operator to notify that the volume is not scratch, but still prompts the operator to allow the initialization to continue.

In a DFSMSrmm environment, the initialization process is different. A utility, EDGINERS, should be used instead of IEHINITT for the tapes defined in the DFSMSrmm CDS. You must ensure that your operational procedures are updated in order to use the new DFSMSrmm utility after the conversion process.

DFSMSrmm does not require that you initialize tape volumes before they are available as scratch, but it prevents you from using as scratch a volume that requires initialization and will not include the volume in a scratch pull list. When adding scratch volumes to DFSMSrmm, you have the choice of deciding whether they need to be initialized or not.

Operators should use the DFSMSrmm EDGINERS utility (instead of IEHINITT) to initialize and erase tape volumes. EDGINERS automatically defines new volumes it initializes or erases to the DFSMSrmm CDS. For an existing volume, DFSMSrmm ensures that the proper volume is mounted and the requested action is actually required. You can request DFSMSrmm to automatically initialize scratch volumes when you add them to DFSMSrmm.

Archived

6.2.10 Tape pool management

TLMS allows you to define ranges of tape volumes in a tape library to a specific pool. It restricts scratch requests to a specific pool based on creating the user and/or data set name. For example, you define ranges of tape volumes dedicated to a specific application use only.

Multiple scratch pool support

TLMS supports multiple scratch pool addressing by using a table called TLMSRTAB where you can specify a data set name and the range of volumes associated with it. If the volume opened for output does not fall within the specified range, TLMS denies the request and dismounts the volume.

If your installation uses the TLMSRTAB table to access different scratch pools, you must use the DFSMSrmm multiple scratch pools facility. With this support, you will be able to direct new tape data sets to specific scratch pools on the basis of job name and data set name. Thus, you can discretely manage sets of volumes on a client, application, or another basis.

DFSMSrmm can manage system managed tapes as well as any other pooling driven by an external product. You can use the DFSMSrmm installation exit EDGUX100 to support multiple scratch pools. The EDGUX100 exit can select a specific scratch pool for nonspecific tape volume output requests by using the data set name and jobname as the selection criteria.

Implementation for DFSMSrmm

DFSMSrmm decides which pool to use on the basis of the VLPOOL definitions in the DFSMSrmm PARMLIB member. This function is dependent on the correct VLPOOL definitions, but you can decide which pool to select either by system by using the SYSID operand of the VLPOOL statement in the EDGRMMxx PARMLIB member, or by using the EDGUX100 installation exit.

DFSMSrmm intercepts the mount messages issued on the system, and, for those for nonspecific volume requests, calls the installation exit to enable it to make a decision. DFSMSrmm saves the scratch pool selected by the exit for use in updating tape drive displays, updating the mount message, and validating the volume when one is mounted.

The exit can optionally request that DFSMSrmm prevent the cartridge loader from being indexed. Thus, drives can be selectively preloaded with scratch volumes from specific pools, but scratch volumes from other pools can be manually mounted without the cartridge loader being emptied.

When the exit selects a pool, the only acceptable volumes are those in that pool. When DFSMSrmm selects the scratch pool without use of the exit, any scratch volume from any eligible scratch pool on that system is acceptable.

DFSMSrmm provides a working sample exit that implements a scratch pooling system based on job names and data set names; all you have to do is update a table containing job names, data set names, and pool prefixes in the EDGUX100 exit.

You can also specify different data set or job name pool tables for each system. If you specify the NAME operand in the VLPOOL command in PARMLIB member EDGRMMxx, the drive will display this name rather than the pool prefix.

Note: The specification of a NAME value allows DFSMSrmm to drive Basic Tape Library Support (BTLS) scratch pool selection. You can use data set names as well as job names to direct scratch pool selection. Use pool NAME values that match the BTLS pool names, such as SCRATCH1.

6.2.11 Security

TLMS provides security options that control online inquiry and update. To change these options or add new users after installation, you must build an SMP USERMOD and then restart TLMS.

DFSMSrmm uses profiles in the FACILITY class of RACF to determine a user's authority to request actions in DFSMSrmm. DFSMSrmm uses profiles with the *STGADMIN.EDG.* prefix. You can control every command or DFSMSrmm function with this integrated RACF interface.

TLMS can use a RACF interface called TLMSRACF, which allows you to automatically delete the tape volume from the RACF database when a tape is scratched.

The RACF tape profile must be already defined.

If in your actual environment you are using the RACF TAPEVOL or TAPEDSN classes to protect tape volumes, you no longer need to define the tape volumes to RACF because DFSMSrmm can do this task automatically.

You must specify either TPRACF(A) option in the DFSMSrmm PARMLIB member, or use TPRACF(P), which allows you to use predefined RACF tape profiles. Otherwise, you must use TPRACF(N).

Additional information can be found in the *DFSMSrmm Implementation and Customization Guide*, SG26-7405.

6.3 Retention and movement

In the sections that follow, we describe some basic tape management retention and movement concepts, and suggest some factors to consider when converting from TLMS to DFSMSrmm.

6.3.1 Retention management

Both TLMS and DFSMSrmm provide comprehensive retention criteria; however, the criteria are defined and processed in different ways.

TLMS policies

Default retention periods can be specified in a user table, TLMSDTAB; this specification allows definition of:

- Minimum days for specific data sets
- Minimum days for groups of data sets
- Installation-wide default for any other data sets

The RMF allows nine different retention policies, which are listed in 5.3, “Retention methods” on page 188.

TLMS allows three types of data set names. Table 6-6 describes them and gives the equivalents in DFSMSrmm.

Table 6-6 TLMS data set name types

	Description	DFSMSrmm equivalent
F	Fully qualified data set name	DSN(‘dsname’)
Q	Qualified data set name ending with a period	DSN(‘dsname.**’)
P	Partially qualified data set name	DSN(‘dsname*.**’)

If the data set type is P or Q and the retention type is 4 (cycle control), the data set name may be a GDG. DFSMSrmm provides more precise control by allowing a type of GDG to be specified along with a GDG base data set name. It checks for the standard version qualifier of GnnnnVnn during VRS processing. Also, a pseudo-GDG allows a customer to define a pattern that DFSMSrmm treats like a GDG.

You must use your knowledge of your installation to decide whether an equivalent VRS definition should use the .** suffix with the GDG or NOGDG operand, because TLMS does not specifically identify GDG data set names.

TLMS allows an additional qualifier, job name, to its selection criteria.

Archived

DFSMSrmm can tailor this selection criteria by using a mask for the job name, and specifying in PARMLIB whether the job name should take precedence over the data set name during VRS match processing.

TLMS allows several combinations of retention types for a single definition; for example, elapsed days retention followed by cycle retention followed by days since created and cataloged. DFSMSrmm provides equivalent function, and many additional retention and movement options.

The EDGCDYNM extract program converts TLMS retention methods to the best equivalent rules in VRS definitions as explained in 5.3, “Retention methods” on page 188.

To avoid VRS definitions with duplicate data set and job names being generated, EDGCDYNM processes only the first occurrence of a data set and job name combination and ignores all others.

A warning message is produced for each ignored data set. These should be reviewed after conversion to ensure that the desired retention criteria is being applied.

In addition, TLMS can use specific or unique Julian dates to control tape data set expiration and retention. These keyword dates can be overridden by the installation's retention policies, which are defined in the RMF:

EXPDT=98000	Specifies that the tape volume being processed is a foreign tape, not under TLMS control
EXPDT=98*ddd*	Specifies that a tape data set be retained until the specified number of days (***ddd***) since last reference have elapsed. This is equivalent to RMF type 8
EXPDT=99000	Specifies that a tape data set be retained for as long as it is cataloged. This is equivalent to RMF type 1.
EXPDT=990*cc*	Specifies that the total number of cycles (***cc***) or generations of the data set be kept. This is equivalent to RMF type 4.
EXPDT=991*dd*	Specifies catalog and date control, where ***dd*** is the number of days since creation. This is equivalent to RMF type 3.
EXPDT=992*dd*	Specifies date control, where ***dd*** is the number of days since creation. This is equivalent to RMF type 2.
EXPDT=99365	Specifies that a tape data set be retained permanently. This is equivalent to RMF type 7.
EXPDT=99366	Specifies that a tape data set be retained permanently. This is equivalent to RMF type 7.

Archived

If a job abends while creating a tape data set, TLMS flags the corresponding DBD record and sets a default retention period of one day. The volume may then be returned to the scratch pool after the next run of TRS. The DFSMSrmm equivalent is to use an ABEND VRS as shown in Example 6-2.

Example 6-2 RMM add an ABEND VRS

```
RMM ADDVRS DSN('ABEND') -
     DAYS COUNT(1) -
     RELEASE(EXPIRYDATEIGNORE)
```

By using VRSs and the EDGUX100 exit, DFSMSrmm supports the JCL special expiration dates honored by other tape management systems. You define VRSs that can be used to provide management function equivalent to the special Julian dates of TLMS. You then assign that VRS to the data set using either a management class or the installation-wide exit to set a VRS management value. See Example 6-3.

Example 6-3 JCL example to use a special EXPDT

```
//MYDSN  DD DSN=MY.DATA,DISP=(,CATLG),LABEL=(,SL,EXPDT=99000),
// additional parameters
```

And then you define the VRS as shown in Example 6-4.

Example 6-4 RMM add a VRS for spezial EXPDT processing

```
RMM ADDVRS DSN('D99000') WHILECATALOG
```

At open time the DFSMSrmm EDGUX100 installation-wide exit requests a VRS management value of D99000, having determined that you specified a “special date” in your JCL.

During inventory management VRS processing, if there is no VRS data set mask covering the data set, DFSMSrmm matches the data set VRS management value with the ADDVRS DSN('D99000') definition, and retains the data set for as long as it is cataloged.

You can list the management classes and VRS management values assigned to data sets using either the RMM LISTDATASET command or the ISPF dialog, or by reading the report extract file data set records.

However, you have three alternatives to gradually convert away from using special dates:

- Using system-managed tape, an ACS routine can assign a management class corresponding to a special date. DFSMSrmm uses this management class to relate to an associated VRS definition to provide the required retention criteria.

- A DFSMSrmm installation-wide exit can provide a VRS management value that is used in a similar way to the management class.
- Use management class RETENTION LIMIT.

Note: The SMS management class takes precedence over the VRS management value.

The VRS retention hierarchy is shown in Figure 6-1.

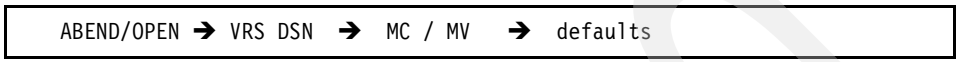


Figure 6-1 VRS hierarchy

While the EDGCDYNM program allows for processing of existing special expiration dates, and the creation of VRS management values and definitions, it is the installation's responsibility to ensure the following:

- Any new retention criteria is defined as VRS management values and VRS definitions.
- Either an appropriate DFSMS ACS routine or DFSMSrmm installation exit EDGUX100 is installed and active to set VRS management values from special expiration dates.

  With APAR OW30969 a new sample of the EDGUX100 installation exit called EDGCVRSX is available in SAMPLIB that uses a dynamic table driven assignment method based on data set names and keyword dates. We recommend that those converting to DFSMSrmm use the EDGCVRSX sample and the table generated by the conversion tools.

DFSMSrmm flags a data set if the job that writes to it leaves the data set open. This can occur only through failure of the MVS system, and not as a result of abends. Open data sets are retained by VRS processing, and the specified or calculated expiration date is kept. Thus, manual intervention might be required to release the volume if you specified a long retention period. Otherwise, you can use the special OPEN VRS retention to establish how to manage this situation. The new VRS option RELEASE(EXPIRYDATEIGNORE) can be used on both ABEND and OPEN VRSs so that any default or specified expiration date is ignored when a job abends or leaves a data set open.

Pay attention to the GDG processing, which sometimes differs between the two products. For example, if you generate a duplicate generation for a data set, TLMS uses both generations with the same name in the count of the GDG versions. DFSMSrmm uses only the latest generation, and releases the other one.

DFSMSrmm policies

DFSMSrmm does not recognize that any expiration dates have special meanings; but, by using the installation-wide exit EDGUX100, or a management class with VRS, you can get DFSMSrmm to provide the same function. DFSMSrmm uses VRSs, which define the policy to control the tape data set retention and tape volume retention. Expiration dates of 99365 and 99366 or 1999/365 and 1999/366 are considered *never-scratch* dates in DFSMSrmm.

DFSMSrmm allows you to set a system default retention period if the retention or expiration is not specified in the JCL during data set and volume creation. You can extend the retention period for the data set on the volume or for the volume, after it has been created using a VRS definition. The retention period specified in the VRS can override the expiration date in the DFSMSrmm CDS.

DFSMSrmm supports many different types of retention. For a data set, you can request the following types of retention using specific or generic data set names:

- *Retention by cycles* or *retention by cycles while cataloged*, which defines how many cycles or generations of a data set should be retained. Cycles can either be by data set or by day, as in CYCLES or BYDAYSCYCLE.
- *Retention by number of elapsed days since creation* or *retention by number of elapsed days since creation while cataloged*, which indicates that a data set be retained until the specified number of days since creation has elapsed.
- *Retention by number of elapsed days since last reference* or *retention by number of elapsed days since last reference while cataloged*, which indicates that a data set be retained until the specified number of days since the last reference has elapsed.
- *Retention by catalog*, which indicates that a data set be retained as long as it is cataloged.
- *Retention by extra days*. On a NEXTVRS you can specify 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. This option can only be used on a NEXTVRS and not a primary VRS.
- *Retention until expired*, which means that a data set can now optionally be managed separately for retention. A vital record specification management value or a SMS management class can identify retention criteria, and the data set name can be used to identify movement criteria. Up to two VRSs can be used to provide policy information for a data set. The first VRS is called the *primary VRS*, and if there is a secondary VRS, it is called the *secondary VRS*

When new data is created, and no vital record specification management value or management class is assigned because the jobs do not specify keyword dates and are not SMS managed, VRSEL processing finds only the data set VRS to match to, which specifies UNTILEXPIRED. DFSMSrmm checks the volume expiration date and the UNTILEXPIRED option makes DFSMSrmm drop the data set from retention.

The volume expiration date is also used if the following conditions are true and the PARMLIB option VRSEL(NEW) is specified:

– When a data set matches to a single VRS
– When UNTILEXPIRED is used in the secondary VRS

The VRS enhancements introduced with APAR OW30969 enable until expired retention to be based on any type of retention. When a primary and secondary VRS match to a data set, the use of UNTILEXPIRED in the primary uses the secondary VRS chain to determine overall retention. If there is no secondary VRS, or UNTILEXPIRED is used in a secondary VRS or a VRS management value VRS, and PARMLIB option VRSEL(NEW) is used, only the volume expiration date is used. If VRSEL(OLD) is specified in PARMLIB, UNTILEXPIRED retention uses the volume expiration date or catalog status to determine for how long to retain a data set.

Any policy used for retention or movement, or both retention and movement, can be a chain of multiple VRSs, where each VRS in a chain can use a different type of retention. Both the primary and secondary VRS can be VRS chains, offering enough flexibility to create quite complex retention and movement policies.

A secondary VRS can be only a VRS management class or a VRS management value. For more information on how the new VRS processing works, refer to *DFSMSrmm Primer,* SG24-5983.

For a volume, you can request the following types of retention using a specific or generic VOLSER:

- *Retention by number of elapsed days since creation*, which indicates that a volume be retained until the specified number of days since creation have elapsed
- *Retention by number of volumes*, which indicates how many volumes should be retained

For all data sets for which you do not specify a retention date in JCL, DFSMSrmm uses a default retention period. You use VRSs to extend this date. For data sets that are left open or that are closed by abend processing, DFSMSrmm provides special ABEND and OPEN VRSs to allow you to control the retention separate from the normal retention for those data sets.

Archive

Figure 6-2 and Figure 6-3 show how to add a data set VRS to set a default retention management period that protects the data set as long as it is cataloged.

```
  Panel  Help
------------------------------------------------------------------------------
                   DFSMSrmm Add Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . . 'SSMSPB.APPLIC.BACK*.**'
       Job name mask . . BACKUP*

  Volume serial . .         ( May be generic )

  VRS name  . . . .
```

Figure 6-2 Define VRS for data set name retention

Archived

```
  Panel  Help
------------------------------------------------------------------------------
                          DFSMSrmm Add Data Set VRS
Command ===>

Data set mask . : 'SSMSPB.APPLIC.BACK*.**'                      GDG . . NO
     Job name mask . : BACK*

Count . . . . 99999              Retention type  . . . . . . CYCLES
                                 While cataloged . . . . . . YES
Delay . . . .     Days           Until expired . . . . . . .

Location  . . . . . . HOME
Number in location
Priority  . . . . . . 0
                                 Release options:
Next VRS in chain . .            Expiry date ignore  . . . .
    Chain using . . .            Scratch immediate . . . . .

Owner . . . . . . SSMSPB
Description . . . user applic testing backups
Delete date . . .             ( YYYY/DDD )

Press ENTER to ADD the VRS, or END command to CANCEL.
```

Figure 6-3 Define VRS for retention while cataloged

DFSMSrmm provides support for the specific and unique Julian dates of 98000-99364 having special meanings. The support is provided through an installation-wide exit, which you can use to interrogate and recognize the dates. When the installation-wide exit detects a data set using the Julian dates, it assigns a VRS management value that identifies the retention management policy. A sample EDGUX100, called EDGCVRSX, is available in SAMPLIB, which uses a dynamic table driven assignment method based on data set names and keyword dates. We recommend that those converting to DFSMSrmm use the EDGCVRSX sample and the table generated by the conversion tools.

When the VRSEL(NEW) PARMLIB option is in use, retention types can be combined to meet complex retention need. A data set can now be retained by a data set name VRS and a VRS management class (assigned through SMS ACS routines) or VRS management value (assigned through installation-wide exit EDGUX100) together. DFSMSrmm processes both primary and secondary VRSs to retain a data set.

DFSMSrmm uses the following criteria to retain the data set:

- If a data set name matches to a data set name VRS. DFSMSrmm will use this VRS as the primary VRS.

- The secondary VRS is based on a matching vital record specification management class or vital record specification management value.
- If a match on data set name is not found, the vital record specification management class or vital record specification management value will be called the primary VRS; that means a vital record specification management class or a vital record specification management value can be either a primary VRS (if no matching data set name VRS is found) or a secondary VRS.

When the VRSEL(OLD) PARMLIB option is in use, during inventory management processing, if a match on data set name is not found, DFSMSrmm checks the data set's management value. If the value exists, DFSMSrmm matches the assigned VRS management value with the defined VRS. If there is a match, DFSMSrmm uses the retention specified in the VRS overriding the expiration from the DFSMSrmm CDS. For example, if you want to support the expiration date of 99000 (retention while cataloged), you need to define a VRS to manage the retention while cataloged. In the installation-wide exit, you intercept the tape data set using the 99000 expiration date, and assign a VRS management value matching the defined VRS.

6.3.2 Retention criteria

This section illustrates a sample conversion of various RMF retention combinations.

This example assumes that the TLMSLOCN statements shown in Figure 6-4 are used to map the TLMS locations to DFSMSrmm equivalents.

```
TLMSLOCN TLMSID=DC RMMLOC=SHELF
TLMSLOCN TLMSID=TL RMMLOC=SHELF
TLMSLOCN TLMSID=CL RMMLOC=SHELF
TLMSLOCN TLMSID=D2 RMMLOC=D2
TLMSLOCN TLMSID=FS RMMLOC=FS
TLMSLOCN TLMSID=OS RMMLOC=OS
```

Figure 6-4 TLMSLOCN statements

Data in Example 6-5 is converted into the TSO RMM commands shown in Example 6-6. Actually, only DFSMSrmm CDS records are generated, but you can use the EDGCNVT program VRSCMDS DD to generate the ADDVRS commands, so that you can easily check what is produced for DFSMSrmm.

Example 6-5 Sample retention selection report for migration

```
**ANY COMPANY**                        TAPE RETENTION SELECTION RECORDS                                    TLMS016    PAGE    1
 CA-DYNAM/TLMS 5.3 9012TL530                                                                               13 OCT 1992 11.50.10


                                                                  JOB-NAME
Q-IND   I------- DATA SET NAME OR QUALIFIER -------I   QUALIFIER    ADD DATE       I---- AUTHORIZATION ----I
    -   ---------------------------------------------   --------    --------       -------------------------
             DATA CENTER LOCATION      1ST OFF-SITE     2ND OFF-SITE     3RD OFF-SITE     4TH OFF-SITE     5TH OFF-SITE
              T  ID   NUM    VER       T  ID   NUM      T  ID   NUM      T  ID   NUM      T  ID   NUM      T  ID   NUM
              -  --  ----   ----      -  --  ----      -  --  ----      -  --  ----      -  --  ----      -  --  ----

        **DEFAULT**                                                    13/03/89       STSG002          00022000
              1  DC
    F   ACSS.DUMP.LOADDATA.D0001                                       27/07/89       SDAC005
              2  DC  0090
    Q   ACSS.DUMP.RAWDATA.                                             22/06/89       SDAC005
              2  DC  0090
    P   ADB2.ARCHLOG                                                   01/06/89       SDAC005
              6  DC                    5  FS  0001      5  TL  0013
    Q   ADB2.IMAGCOPY.                                                 18/05/89       SDAC005
              6  DC                    5  FS  0001      5  TL  0020
    Q   PCSS.DFDSS.                                                    30/01/90       SDAC018
              6  DC                    4  FS  0002      1  TL
    Q   STSG.VTAM.BACKUP.                                              27/06/90       STSG001
              6  DC                    4  FS  0002      4  D2  0005      1  TL
```

Example 6-6 Sample converted VRS definitions

```
ADDVRS DSN('**') WHILECATALOG LOC(HOME) COUNT(99999) OWNER(STSG002)
ADDVRS DSN('ACSS.DUMP.LOADDATA.D0001') DAYS LOC(HOME) COUNT(90)       -
       OWNER(SDAC005)
ADDVRS DSN('ACSS.DUMP.RAWDATA.**') DAYS LOC(HOME) COUNT(90)           -
        OWNER(SDAC005)
ADDVRS DSN('ADB2.ARCHLOG*.**') DAYS LOC(FS) COUNT(1) OWNER(SDAC005)   -
       STORENUMBER(1) NEXTVRS(N1)
ADDVRS NAME(N1) DAYS COUNT(13) LOCATION(HOME) STORENUMER(13)          -
       OWNER(SDAC005)
ADDVRS DSN('ADB2.IMAGCOPY.**') DAYS LOCATION(FS) COUNT(1)             -
       OWNER(SDAC005) STORENUMBER(1) NEXTVRS(N2)
ADDVRS NAME(N2) DAYS COUNT(20) LOCATION(HOME) OWNER(SDAC005)          -
       STORENUMBER(20)
ADDVRS DSN('PCSS.DFDSS') GDG CYCLES LOCATION(FS) COUNT(2)             -
       OWNER(SDAC018) STORENUMBER(2) NEXTVRS(N3)
ADDVRS NAME(N3) WHILECATALOG LOCATION(HOME) OWNER(SDAC018)
ADDVRS DSN('STSG.VTAM.BACKUP.**') CYCLES LOCATION(FS) COUNT(2)       -
       OWNER(STSG001) STORENUMBER(2) NEXTVRS(VRSSTSG1)
ADDVRS NAME(VRSSTSG1) OWNER(STSG001) COUNT(5) CYCLES LOCATION(D2)     -
       STORENUMBER(5) NEXTVRS(VRSSTSG2)
ADDVRS NAME(VRSSTSG2) OWNER(STSG001) WHILECATALOG LOCATION(HOME)
```

Duplicate RMF definitions

To avoid VRS definitions with duplicate data set and job names being generated, EDGCDYNM only processes the *first* occurrence of a data set and job name combinations and ignores all others.

Archived

A warning message is produced for each ignored data set. These should be reviewed after conversion to ensure the desired retention criteria are being applied.

Non-GDG processing

TLMS allows the specification of full or qualified data set names, whereas DFSMSrmm allows full generic specification qualified by GDG or non-GDG.

The conversion process produces VRS definitions for non-GDG data sets. These should be reviewed after conversion for any data sets that may be GDGs. This is particularly important if CYCLES control is being used. A NOGDG VRS definition can retain GDG data sets, but only if the data set mask is generic and includes a generic character at the end (as shown in Figure 6-5) that covers the last qualifier of the GDG data set.

Q A.B.	is translated to	DSN('A.B.**')

Figure 6-5 VRS data set mask conversion

This matches to both 'A.B.C' a non-GDG, and 'A.B.G0010V00' a GDG.

Multi-data set multi-volume retention

DFSMSrmm can return a single volume in a multi-volume set of volumes to scratch; it does not need to operate always on the complete set of volumes. Consequently, you may see DFSMSrmm return a volume to scratch that is still retained by TLMS.

Each data set in a multi-volume set can be retained individually using VRS. If there are data sets in the volume set that have different retention requirements, DFSMSrmm can attempt to release them earlier. For a volume to be released, *all* data sets on that volume must no longer be retained by VRS.

If you use a controlling file to cause the complete set to be retained, you may require additional VRS definitions under DFSMSrmm to cause the same volumes to be retained. DFSMSrmm includes PARMLIB options to make RMM work in a similar way as TLMS. Use the MOVEBY(SET) and RETAINBY(SET) options to manage multi-volume sets as an aggregate. Alternatively, you can benefit from DFSMSrmm releasing volumes early when the data is not required to be retained.

Archived

6.3.3 Storage location management

You define and use your installation's management policies for those tape volumes and data sets that contain installation-critical data for disaster recovery. These policies define:

- Critical data: Data that is vital to your installation
- Storage location: Where to store critical data
- Retention criteria: How long to keep the critical data in the storage location
- Movement criteria: How to move critical data to and from the data center and the storage locations.

Because DFSMSrmm and TLMS differ in terms of how they define and process storage location management policies, it is important that you review the current storage location management policy to understand retention and movement criteria.

TLMS

TLMS supports up to five storage locations. Each location can be one of three types of storage:

- Undefined: Usually standard racking as in the tape library
- Cabinets; One or more cabinets, each containing slots. These normally correspond to fireproof safes.
- Boxes: These normally be located in an offsite store.

TLMS manages the movement of tape volumes between locations through a batch program that checks on your installation-defined storage location policies. These policies are based on data set name, fully or partially qualified, and are further qualified by job name or owner. The storage location policies are defined as part of the retention policies in the RMF.

TLMS automatically checks tape volumes meeting the criteria and assigns the appropriate storage location and slot or box number. In addition, it generates a report indicating the current location, the destination location, and the slot or box numbers for tape volumes that require movement between locations.

DFSMSrmm

Using your installation-defined VRSs, DFSMSrmm automatically controls the movement of the data sets on a tape volume or the tape volume from the tape library to and among the storage locations. You can specify a data set name or a data set name mask and, optionally, a job name or job name mask to define a VRS. Additionally, you can specify a VOLSER or a generic VOLSER.

Archived

Figure 6-6 and Figure 6-7 show how to add a data set VRS defining the storage location and the retention period to retain one copy for as long as the data set is cataloged.

```
 Panel  Help
------------------------------------------------------------------------------
                  DFSMSrmm Add Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . . 'MY.CRITICAL.DATA'
       Job name mask . . D094576

  Volume serial . .         ( May be generic )

  VRS name  . . . .
```

Figure 6-6 Define VRS for data set name movement

Archived

```
 Panel  Help
------------------------------------------------------------------------------
                          DFSMSrmm Add Data Set VRS
Command ===>

Data set mask . : 'MY.CRITICAL.DATA'                              GDG . . NO
     Job name mask . : DO94576

Count . . . . 5                 Retention type  . . . . . . CYCLES
                                While cataloged . . . . . . YES
Delay . . . .     Days          Until expired . . . . . . . NO

Location  . . . . . . OFFSITE
Number in location    5
Priority  . . . . . . 0
                                Release options:
Next VRS in chain . . NXT1      Expiry date ignore  . . . . YES
    Chain using . . . AND       Scratch immediate . . . . . YES

Owner . . . . . . SSMSPB
Description . . .
Delete date . . . 1999/365    ( YYYY/DDD )

Press ENTER to ADD the VRS, or END command to CANCEL.
```

Figure 6-7 Define VRS for application movement policy

The meaning of the fields **Chain using . . . AND** and **Release options:** is the same as the following subparameters used on the TSO RMM ADDVRS subcommand:

ANDVRS Specify ANDVRS to create a chain of VRS where all the retention conditions must be true for the data set to be retained. DFSMSrmm uses the STORENUMBER and LOCATION from the first VRS in the ANDVRS subchain.

RELEASE Specifies special release options after a data set is released. RELEASE options take affect only for data sets retained by a VRS:

- EXPIRYDATEIGNORE is specified so DFSMSrmm ignores the volume expiration date when VRSEL(NEW) is used and no other vital VRS matches to the volume.
- SCRATCHIMMEDIATE is specified so DFSMSrmm returns the volume to scratch status in a single run of inventory management EXPROC processing when the only pending release action for the volume is to return to scratch.

Archived

DFSMSrmm does not restrict the movement of the data set on the tape volume or the tape volume to and among the storage locations. When a data set on the tape volume, or the tape volume is moved from a tape library to and among the storage locations for disaster recovery or vital records, DFSMSrmm assigns a storage location bin number, storage location name, and store date.

DFSMSrmm allows you to specify one location name in each data set or volume VRS. If you require that a data set or volume be moved to more storage locations, you can chain together two or more VRSs by using the NEXTVRS, NAME, and ANDVRS keywords. DFSMSrmm does not restrict any sequence of data set or volume moves among storage locations and system-managed libraries. For example, if your installation's policy is to initially retain the data set in the *remote* storage location, and then later retain it in the *distant* storage location, you first define a data set VRS for the *remote* storage location and indicate that the data set requires multiple moves by specifying a name VRS. You then indicate the next storage location, the *distant* storage location, through that name VRS.

Figure 6-8 and Figure 6-9 show how to add a data set VRS defining the first storage location (remote), the retention (one cycle), and the name of the next storage location.

```
 Panel  Help
_______________________________________________________________________________
                    DFSMSrmm Add Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . . 'MY.CRITICAL.DATA'
       Job name mask . . D094576

  Volume serial . .          ( May be generic )

  VRS name  . . . .
```

Figure 6-8 Define VRS for GDG data set name

Archive

```
  Panel  Help
------------------------------------------------------------------------------
                          DFSMSrmm Add Data Set VRS
Command ===>

Data set mask . : 'MY.CRITICAL.DATA'                              GDG . . NO
     Job name mask . : DO94576

Count . . . . 5                 Retention type  . . . . . . CYCLES
                                While cataloged . . . . . . YES
Delay . . . .     Days          Until expired . . . . . . . NO

Location  . . . . . . REMOTE
Number in location    12
Priority  . . . . . . 0
                                Release options:
Next VRS in chain . . NEXTOO    Expiry date ignore  . . . . YES
    Chain using . . . NEXT      Scratch immediate . . . . . YES

Owner . . . . . . SSMSPB
Description . . .
Delete date . . . 1999/365    ( YYYY/DDD )

Press ENTER to ADD the VRS, or END command to CANCEL.
```

Figure 6-9 Define VRS for application first store

Figure 6-10 and Figure 6-11 show how to add a name VRS defining the second storage location (*distant*) and the retention period (one cycle). This VRS is chained by the VRS definition in Figure 6-9.

```
  Panel  Help
------------------------------------------------------------------------------
                    DFSMSrmm Add Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . .
       Job name mask . .

  Volume serial . .          ( May be generic )

  VRS name  . . . . NEXTOO
```

Figure 6-10 Define name VRS

```
  Panel  Help
_____________________________________________________________________________
                            DFSMSrmm Add Name VRS
Command ===>

Name  . . . . . : NEXT00


Count . . . . 1                 Retention type  . . . . . . cycles
                                While cataloged . . . . . . no
                                Until expired . . . . . . . no

Location  . . . . . . DISTANT
Number in location    2


Next VRS in chain . .
    Chain using . . .

Owner . . . . . . SSMSPB
Description . . .
Delete date . . .             ( YYYY/DDD )

Press ENTER to ADD the VRS, or END command to CANCEL.
```

Figure 6-11 Define VRS for application second store

During inventory management processing, DFSMSrmm identifies the volumes that must be moved to and among the storage locations, and assigns the exact store bin numbers where the volumes are to be stored. DFSMSrmm provides a batch utility (EDGRPTD), which you can use to generate a pull list for the volumes requiring movement. When the volumes have been moved, you must confirm that the moves have taken place. The confirmation process updates the DFSMSrmm CDS to reflect the most current information about where the volumes are stored.

DFSMSrmm uses the retention criteria from the VRS to manage tape volume retention in the storage locations. Refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405, for more information about DFSMSrmm retention management.

Note: Each time you run inventory management VRSEL processing, DFSMSrmm recalculates retention and movement needs. If you delete a VRS, any data sets it retained can match to another VRS or are eligible for release processing.

Archived

DFSMSrmm supports an almost unlimited number of storage locations. You should be able to convert your TLMS storage locations to DFSMSrmm without hitting a limit caused by their number. In DFSMSrmm there are three built-in storage locations that are all managed by bin number (slot), but by using LOCDEF commands in the DFSMSrmm PARMLIB you can use installation defined names, including those you are familiar with under TLMS, and decide whether to have them shelf managed or not.

When you convert to DFSMSrmm you have the choice to remain using the storage location names you currently use with TLMS, or use different names. You may wish to use different names because DFSMSrmm allows up to eight character names rather than just two characters. If you decide to use the DFSMSrmm built-in storage location, LOCAL, DISTANT, and REMOTE, you do have to translate location names and restricted to three storage locations.

You can use a combination of built-in and installation defined storage locations requiring some translation of names. When you decide to use different names under DFSMSrmm, you can translate the TLMS storage location names to DFSMSrmm storage location names either during the EDGCDYNM run, or during the EDGCNVT run. To translate the names with EDGCDYNM, use the TLMSLOCN RMMLOC keyword to specify a different name. To translate the names with EDGCNVT, use the LOCDEF SYSIN statement and the IF STORLOC statement to specify a different name.

DFSMSrmm also supports an almost unlimited number of loan locations. If you wish to use these as a replacement for any of the TLMS storage locations, you can do so by using the LOCDEF statement in EDGCNVT to identify the location as a type of LOAN. When running EDGCDYNM, you should treat it as a storage location, and use TLMSLOCN and DFRMMLOC statements.

6.3.4 Movement policies

DFSMSrmm uses your installation-defined VRSs (location and retention management policies) to control the movement of tape volumes between the storage locations that you have defined.

To define a VRS, you can use:

- A data set name or data set mask
- A job name or job mask
- A VOLSER number or VOLSER mask
- The management class or VRS management value

When a tape volume is to be moved from a tape library to a remote storage location for disaster recovery, DFSMSrmm assigns a store bin (slot) number, a store ID (location name), and a store date.

This target location does not become the current location of the volume until confirmed by the operator or tape librarian.

Further volume movement of a volume is prohibited until the existing movement is confirmed.

Movement confirmation can be for one of the following:

- Individual volume
- All volumes from one location to another
- All volumes from one location
- All volumes to a location
- All volumes ready to be scratched
- All volumes not ready to be scratched
- Any combination of the above

All volumes defined to DFSMSrmm have a rack number assigned to provide shelf management while in the data center, and a unique external identifier for the volume. EDGCDYNM and EDGCNVT use the TLMS VOLSER as the DFSMSrmm rack number.

Shelf management in DFSMSrmm storage locations is optional, but if provided, it is done using bin numbers. Each bin number represents a single slot in which to store a volume. The EDGCDYNM OPTIONS statement lets you control what the conversion process does for bin numbers. You know, from the TLMS015 report, whether your storage locations are managed as cabinets or boxes, or are undefined storage. You must decide how best to convert the information to DFSMSrmm. Both the VOLLOCN=PASS and VOLLOCN=PROCBIN options pass on bin information; PASS passes on the cabinet and box numbers, and PROCBIN generates bin numbers starting from 000001. Here are some options:

- Specify VOLLOCN=PROCLOC and use DFSMSrmm inventory management to assign bin numbers and locations.
- Use VOLLOCN=PROCBIN and have bin numbers generated for you. These are passed from EDGCDYNM to EDGCNVT, and, although they are simply numeric bin numbers starting from 000001 through 999999, you can use them either with the built-in or installation-defined storage locations. If you decide on non-shelf-managed storage locations, EDGCNVT simply ignores the numbers.
- Specify VOLLOCN=PASS and use the existing cabinet and box numbers under DFSMSrmm. Use of this option requires you to make some changes to the sample extract and conversion programs, as the cabinet and box numbers are only four digits, and DFSMSrmm requires six digits.

Cabinet numbers can be expanded to six digits (see Figure 6-12), and used as bin numbers with installation-defined storage locations.

```
CCnn could be expanded to either
CC0Onn or
CCnnOO
```

Figure 6-12 Cabinet expanding until conversion

You can then define bin numbers as CC0001 through CC0099, or CC0100 to CC9900, always leaving the last two digits as zero (00). A small change to EDGCDYNM just before label PVMFS120 can insert 00 into the cabinet number. DFSMSrmm shelf management is very similar to the cabinet management provided by TLMS. If the location is NL and the cabinet number is AA23, the DFSMSrmm bin number becomes NLAA23. This is a value that can readily be identified by your operations staff. If you would rather use a different approach for converting cabinet numbers or you use box numbers, you must modify the sample code.

Box slot numbers are not kept in the DBD, but you can process them in a similar way to the cabinet numbers, and generate the last two digits in numerical sequence for each volume found in the same box as shown in Figure 6-13.

```
BBBB could be expanded to
BBBBnn
```

Figure 6-13 Box slot number expanding until conversion

You can then define bin numbers BBBB01 through BBBB99 to DFSMSrmm. Be sure to define the location as an installation-defined storage location by using LOCDEF. Remember that DFSMSrmm does not provide box management, and can only use the box and slot numbers as conventional open shelving.

6.4 Interfaces

Before the conversion process you must consider the interfaces between DFSMSrmm and other products. Some of these interfaces may be in place on your current TLMS environment.

6.4.1 SMS interface

If you have an IBM 3494 or IBM 3495 Automated Tape Library Dataserver, you must ensure that it works correctly in the DFSMSrmm environment. You have to change the OAM exits using the DFSMSrmm version, and provide the correct input to the extract program for those volumes that belong to the tape library.

You also must define a DFSMSrmm location with the same name as the tape library name, using a location type of ATL. Thus, the TLMS location name that you are using for the ATL tapes must be translated to this DFSMSrmm location.

6.4.2 BTLS interface

If your installation uses BTLS to drive IBM 3494 and IBM 3495 Automated Tape Library Dataserver operations, you will need to notify BTLS when a volume has to return to SCRATCH status. In a TLMS environment this is done with a batch job that gets information from the TLMS database and then updates the BTLS catalog. In DFSMSrmm you can use a simple REXX CLIST that synchronizes the DFSMSrmm database with the BTLS catalog.

For a sample CLIST, refer to the redbook *DFSMSrmm Primer,* SG24-5983.

6.4.3 Short-on-scratch processing interface

DFSMSrmm offers an integrated SMS interface that updates the TCDB automatically.

In addition, without manual intervention, you can free volumes that are in PENDING RELEASE status when a short-on-scratch condition is detected inside the ATL. Use the SCRATCHPROC(RMMSCR) PARMLIB option, where RMMSCR is the name of the procedure where DFSMSrmm starts to replenish scratch volumes in the ATL.

6.4.4 TSO and ISPF interfaces

DFSMSrmm has a simple TSO interface that uses a set of powerful commands to drive all tape activities.

There is also an ISPF interface that offers menus for end users, librarians, administrators, and support personnel. You can choose a full DFSMSrmm dialog or a local defined dialog. You can tailor the correct ISPF dialog for each user who will have access to DFSMSrmm.

Archive

6.4.5 Other interfaces

TLMS provides an online transaction that accesses CICS, TSO, ISPF, and ROSCOE environments. You can use inquiry and update processing through one of these interfaces, or you can use an operator console to issue TLMS commands.

DFSMSrmm can use any TSO RMM subcommand in batch mode using the IKJEFT01 or IKJEFT1A TSO batch interface. It can also process any DFSMSrmm command through the API. To use the DFSMSrmm API you need to code in the High Level Assembler language. A preferred method is to use the DFSMSrmm commands from a REXX environment where you can retrieve information back as variables.

If you have any of these interfaces running in your TLMS tape environment, you must change them to reflect the equivalent function in DFSMSrmm.

6.5 Sample extraction programs

In this section we describe the sample programs and JCL for extracting data for the conversion from TLMS to DFSMSrmm. The samples are as supplied in the SAMPLIB library. This book describes them as they were when the book was written. Before running the extract program, you should read and be familiar with the latest version of the document *Data Extraction Programs User's Guide,* supplied as EDGCMM01 member in SAMPLIB, as it will be maintained to reflect changes and enhancements.

Table 6-7 lists the samples for the extraction of data from TLMS.

Table 6-7 IBM-provided extraction samples for TLMS

Name	Location	Function
EDGCDYNM	SAMPLIB	Source for TLMS extract
EDGCMM01	SAMPLIB	Source and formatted user's guide
EDGJDYNM	SAMPLIB	JCL to execute EDGCDYNM

The sample JCL is set up to work in a "typical" installation. Some of the statements in this sample JCL is commented out so that you choose to use it or not, as is your preference. Procedures, which are normally part of other IBM products, are used to assemble and compile the sample code. Thus, you do not have to assemble, link, and compile code before you can use it. If these

Archived

procedures are not available to you, you can edit the JCL to uncomment the statements that execute the programs from a STEPLIB. If you choose to link-edit the programs yourself, you can use the standard link-edit options. None of the programs is re-entrant.

6.6 TLMS data extraction

The data extraction programs are designed to extract relevant information from the TLMS DBD and RMF TLMS016 report (Retention Management File report). The information is used to create four sequential output files containing a series of records that are input to the EDGCNVT program before they are loaded into the DFSMSrmm CDS (see Table 6-8).

6.6.1 Preparing to run the extract program

Before you begin the extraction process, you should consider verifying the integrity of the TLMS DBD to check for broken volume chains, expiration dates in error, or any other condition that can cause an error during data extraction.

You must run the appropriate TLMS utilities to verify the integrity of the database and correct the errors. The TLMS volume chaining verification system (TLMSVCVS) is a monitoring system that checks the validity of the links between volume base records. It explains the errors found in the DBD, and the actions taken for every error message.

Once you are sure that you have a valid DBD, you can proceed to the next step.

6.6.2 The EDGCDYNM program

This section describes the setup and control of the TLMS EDGCDYNM data extraction program.

EDGCDYNM supports all current releases of TLMS. However, the correct level of TLMS must be specified at the time you assemble the sample source code. You use the *SYSPARM* assembler option to specify:

- **TLMS53** - for TLMS Release 5.3, or
- **TLMS54** - for TLMS Release 5.4, or
- **TLMS55** - for TLMS Release 5.5

See Example 6-10 on page 254 for an example JCL. If you do not specify a SYSPARM value, the code is assembled for TLMS 5.4.

Archived

Input files

You need the following DD statements for the input files:

SYSIN Contains optional input parameters to control the execution of EDGCDYNM and the data extraction process

TLMSVMF Defines the TLMS DBD, which contains the tape volume and data set information to be converted

TLMS016 Defines the TLMS Retention Selection Report file (TLMS report 016) which contains the volume retention and movement rules for converting to DFSMSrmm VRS definitions

In addition, you need TLMS REPORT015, which lists the storage locations you use with TLMS. You will use the information in the report to define the TLMSLOCN and DFRMMLOC statements in the SYSIN data set.

Output files

The output files produced by the EDGCDYNM are shown in Table 6-8.

The LRECL of the output files is overridden during program execution. The value included in the JCL examples is just an indication of the approximate record length.

Table 6-8 EDGCDYNM output data sets and records

ddname	Output description	Output record
DEXTLIBD	► Data sets and volumes	► D (data set) record ► L (volume) record
DEXTRBIN	► Empty bin numbers in the storage locations and empty rack numbers in the tape library	► E (empty rack/bin) record
DEXTVRS	► Retention and storage location management policies	► K (VRS) record
DEXTOWN	► VRS and default owner information	► O (owner) record

Note: that the data set and volume (D-type and L-type) records are sequence dependent; see Chapter 7, “Building the CDS” on page 263 for details.

For producing these files, you need to specify the following DD statements:

SYSPRINT Contains informational and error messages associated with the data extraction process

DEXTLIBD Library and data set records for input to the EDGCNVT program

DEXTOWN Owner records for input to the EDGCNVT program

Archived

DEXTRBIN Empty rack and bin records for input to the EDGCNVT program

DEXTVRS VRS records for input to the EDGCNVT program

Input parameters

The EDGCDYNM program accepts a number of optional control statements input in the SYSIN file. These statements are identified by a keyword starting in position 1, followed by one or more parameters starting in position 9. The statements are fixed format; keywords and operands must start in the indicated positions. Optional keywords and operands can be left blank.

Statements can be supplied in any order. Comment statements are identified by an asterisk (*) in position 1.

If you do not specify the control statements in the exact format, the EDGCDYNM program attempts to use the data you enter, and results can be unpredictable, including ABEND0C7, and ignoring of bin counts.

Processing options

The OPTIONS statement is used to control the way in which EDGCDYNM processes the TLMS DBD and RMF data.

Figure 6-14 shows the format of the statement.

```
OPTIONS  OUTSERV=oooooooo VOLCHAIN=ccccccc VOLLOCN=lllllll
*                *        *        *               *
1                18                36              52
```

Figure 6-14 EDGCDYNM SYSIN: OPTIONS control statement

In this statement:

oooooooo Specifies the processing for TLMS out-of-service volumes:

- **Ignore** Specifies that out-of-service volumes are ignored and only empty rack records will be created
- **Process** Specifies that out-of-service volumes are processed as standard scratch or non-scratch volumes
- ***loanloc*** Specifies a 1 to 8 character loan location name. Out-of-service volumes are processed as scratch or non-scratch, and have their loan location set to the specified value.

ccccccc Specifies the processing for volume chains:

Archived

Ignore Specifies that EDGCDYNM process the DBD sequentially and ignore any volume chaining information. All volumes will have a volume sequence number and count of 1.

This option ensures that all volumes are converted, but it loses all multi-volume count and chaining information.

Process Specifies that EDGCDYNM attempts to process volume chaining information to produce correct volume sequencing and counts

Any errors in the DBD volume records may result in missing volumes in the converted data.

/////// Specifies the processing for the volume's home and current location:

Pass Specifies that EDGCDYNM does not convert the volume's current TLMS location. The DFSMSrmm current location and bin number are set to the TLMS location and cabinet or box number. Similarly, the TLMS RMF locations are forwarded into the corresponding DFSMSrmm VRS definitions.

The EDGCNVT program can be used to convert the TLMS information to DFSMSrmm data.

If TLMSLOCN statements are not supplied, the default processing option is PASS.

Procloc Specifies that EDGCDYNM ignore the volume's current TLMS location and sets an DFSMSrmm home and current location of SHELF. The TLMS RMF locations are converted according to the TLMSLOCN specifications.

The DFSMSrmm housekeeping processes, VRSEL and DSTORE, are used after conversion to assign current locations and bin numbers according to the converted VRS definitions.

If TLMSLOCN statements are supplied, the default processing option is procloc.

Procbin Specifies that EDGCDYNM convert the TLMS locations according to the TLMSLOCN specifications. For those volumes in offsite locations, EDGCDYNM assigns a bin number. The

Archived

bin numbers start at 1 for each location, and are incremented by 1 for each volume assigned to that location.

Only one OPTIONS statement should be supplied; if not, the last statement encountered in the SYSIN file is used.

The default values for the OPTIONS statement are shown in Example 6-7.

Example 6-7 EDGCDYNM OPTIONS statement

```
   OPTIONS  OUTSERV=IGNORE   VOLCHAIN=PROCESS VOLLOCN=PROCLOC
```

VRS name prefix

TLMS retention management policies can define multiple volume movements between storage locations. The equivalent DFSMSrmm function requires the use of VRS name definitions to chain together the separate movement definitions. The EDGCDYNM extraction process automatically generates these VRS names according to an installation-defined prefix concatenated with an incrementing number to make it unique.

Figure 6-15 shows the format of the statement.

```
   VRSNAMEP PREFIX=ppppp
   *              *
   1              17
```

Figure 6-15 EDGCDYNM SYSIN: VRSNAMEP control statement

In this statement:

ppppp Is a 1 to 5 character alphanumeric field beginning with an alphabetic.

The generated VRS NAME is extended to eight characters by appending a unique numeric portion.

If a VRSNAMEP statement is not supplied, a default prefix of VRSN is used.

Only one VRSNAMEP statement should be specified; if not, the last statement encountered is used.

Archive

VRS management value prefix

TLMS allows the specification of retention management policies through the use of special expiration dates. The equivalent DFSMSrmm function requires the use of VRS management values and user exit EDGUX100. The EDGCDYNM extraction process optionally can automatically generate these VRS management values according to an installation-defined prefix concatenated with the special Julian date. It also generates the corresponding VRS definition.

Figure 6-16 shows the format of the statement.

```
VRSMGMTP PREFIX=ppp
*               *
1               17
```

Figure 6-16 EDGCDYNM SYSIN: VRSMGMTP control statement

In this statement:

ppp Is a 1 to 3 character alphanumeric field beginning with an alphabetic letter.

If a VRSMGMTP statement is not supplied, special Julian dates are not processed, and VRS management values or definitions are not produced.

Only one VRSMGMTP statement should be specified; if not, the last statement encountered is used. For example, consider the control statement shown in Example 6-8.

Example 6-8 EDGCDYNM VRSMGMTP statement

```
VRSMGMTP PREFIX=VDT
```

This produces (with a volume whose JCL expiry date was 98023) a VRS management value shown in Figure 6-17.

```
VDT98023
```

Figure 6-17 DFSMSrmm VRS management value

Example 6-9 shows how you can add a VRS definition by using the DFSMSrmm TSO subcommand.

Archived

Example 6-9 Add a VRS definition using TSO subcommands

```
ADDVRS DSN('VDT98023') LASTREFERENCEDAYS COUNT(23)
```

Note: The conversion process can only generate management values and VRS definitions for those special dates already in the DBD. For continued use of special dates, the DFSMSrmm user exit EDGUX100 (or an appropriate ACS routine) must be used to set the management value for new tape data sets, and the associated VRS definitions must exist.

Storage location mapping

The TLMSLOCN control statement optionally specifies the mapping between TLMS location IDs and DFSMSrmm storage locations.

Figure 6-18 shows the format of the statement.

```
TLMSLOCN TLMSID=tt RMMLOC=llllllll
*              *       *
1              17      27
```

Figure 6-18 EDGCDYNM SYSIN: TLMSLOCN control statement

In this statement:

tt Is the TLMS storage location ID

llllllll Is the corresponding DFSMSrmm location name. Specify any 1 to 8 character name that is the location name you plan to use under DFSMSrmm. You can specify the TLMS location ID and optionally use EDGCNVT to translate it to the value you want to use.

One TLMSLOCN statement should be supplied for each TLMS location ID to be translated.

Figure 6-19 illustrates a typical mapping.

```
TLMSLOCN TLMSID=TL RMMLOC=SHELF
TLMSLOCN TLMSID=DC RMMLOC=SHELF
TLMSLOCN TLMSID=CL RMMLOC=SHELF
TLMSLOCN TLMSID=D2 RMMLOC=D2
TLMSLOCN TLMSID=FS RMMLOC=FS
TLMSLOCN TLMSID=OS RMMLOC=OS
```

Figure 6-19 TLMSLOCN mapping

Archived

Storage location values

The DFRMMLOC control statement optionally defines a DFSMSrmm storage location. Its principal use is to define system-managed tape libraries (automated and manual) and the maximum number of bins to be used for DFSMSrmm storage locations (local, distant, and remote).

Figure 6-20 shows the format of the statement.

```
DFRMMLOC RMMLOC=llllllll RMMTYP=rrrrrrrr BINS=nnnnnn
*               *               *             *
1               17              33            47
```

Figure 6-20 EDGCDYNM SYSIN: DFRMMLOC control statement

In this statement:

llllllll Is the DFSMSrmm location name. Specify any 1to 8 character name that is the location name you plan to use under DFSMSrmm. You can specify the TLMS location ID and optionally use EDGCNVT to convert it to the value you want to use.

rrrrrrrr Is the type of DFSMSrmm location, one of SHELF, STORE, MTL, or ATL.

nnnnnn Is the six-digit number, from 0 to 999999, of storage bin numbers (slots) in the DFSMSrmm storage location.

It should be specified only for shelf-managed storage locations. You do not need to specify a BINS value as EDGCDYNM generates its own numbers as required starting from 000001. Rack numbers are automatically generated for all volumes no matter where the volume is located.

If DFRMMLOC statements are not supplied, only the basic DFSMSrmm locations shelf, local, distant, and remote are recognized, and no empty bin numbers are generated for the DFSMSrmm storage locations.

Figure 6-21 illustrates a typical mapping.

```
DFRMMLOC RMMLOC=FS       RMMTYP=STORE    BINS=001000
DFRMMLOC RMMLOC=D2       RMMTYP=STORE    BINS=000500
DFRMMLOC RMMLOC=OS       RMMTYP=STORE    BINS=
DFRMMLOC RMMLOC=ROBBY    RMMTYP=ATL
```

Figure 6-21 EDGCDYNM DFRMMLOC mapping

Archived

Volume media names

The VOLMEDIA control statement optionally specifies a volume prefix and the corresponding media name, type, and home location. If VOLMEDIA statements are not supplied, a default media name of 3480 is assigned to all volumes.

Prefixes are scanned in the order of the control statements; to ensure correct processing, the most restrictive prefixes should be specified first.

Figure 6-22 shows the format of the statement:

```
VOLMEDIA VOLPFX=vvvvvv VMEDIA=mmmmmmmm VMTYPE=****t HOMELOC=llllllll
*              *            *              *            *
1              17           31             47           61
```

Figure 6-22 EDGCDYNM SYSIN: VOLMEDIA control statement

In this statement:

vvvvvv Is the 1 to 6 character VOLSER prefix.

mmmmmmmm Is the corresponding DFSMSrmm media name. This should match the media names you specify on the VLPOOL and LOCDEF commands in the DFSMSrmm PARMLIB member.

*******t*** Is the optional media type. It can be any of these:

REEL	Non-cartridge-type tape reel
3480	3480 type cartridge (18 track)
CST	3490 Cartridge System Tape (36 track)
ECCST	3490 Enhanced Capacity Cartridge System Tape (36 track)
HPCT	3590 High Performance Cartridge Tape
EHPCT	3590 Extended High Performance Cartridge Tape

And, if specified, it overrides any values obtained from the TLMS DBD records.

llllllll Is the optional home location name. If specified, this location name must match one of the DFSMSrmm location names, SHELF or any system-managed library name. Storage locations cannot be specified for a home location — EDGCNVT ignores them.
The default home location is SHELF.

Note: The VMEDIA names must match the DFSMSrmm VLPOOL definitions in the EDGRMMxx member of PARMLIB. Otherwise, errors will occur when DFSMSrmm is active.

Archived

Skip processing VOLSERs

The SKIPVOL control statement optionally provides a means to skip processing one or more VOLSERs in the TLMS DBD. This skipping might be required if, for example, there is an error that cannot be corrected in the TLMS data. The skipped volumes can then be manually added after conversion.

Figure 6-23 shows the format of the statement.

```
   SKIPVOL  VOLSER=vvvvvv
   *                *
   1                17
OR:
   SKIPVOL  VOLRNG=vvvvvv-vvvvvv
   *                *       *
   1                17      24
OR:
   SKIPVOL  VOLPFX=pppppp
   *                *
   1                17
```

Figure 6-23 SKIPVOL control statement

In this statement:

VOLSER Specifies a single VOLSER
VOLRNG Specifies a range of VOLSERs
VOLPFX Specifies a VOLSER prefix
vvvvvv Is a six-character VOLSER
pppppp Is a one- to six-character VOLSER prefix

Default volume owner ID

DFSMSrmm requires that all non-scratch volumes be assigned an owner, represented by a 1 to 8 character owner identifier. Because this function is not present in TLMS, a default owner ID must be provided. The DEFOWNER statement lets you provide an owner ID different from the default of LBRARIAN.

Figure 6-24 shows the format of the statement.

```
   DEFOWNER oooooooo
   *        *
   1        10
```

Figure 6-24 EDGCDYNM SYSIN: DEFOWNER control statement

Archived

In this statement:

oooooooo Is a 1 to 8 character alphanumeric field beginning with an alphabetic letter.

If a DEFOWNER statement is not supplied, a default owner of LBRARIAN is used.

Only one DEFOWNER statement should be specified; if not, the last statement encountered is used.

VRS owner ID

DFSMSrmm requires that all VRS definitions be assigned an owner, represented by a 1 to 8 character owner identifier. TLMS provides a 25-character authorization field in the RMF report. EDGCDYNM can attempt to use this field by extracting the first eight non blank characters to form an owner ID. This ID may not always be suitable or valid, so the VRSOWNER statement gives the option of providing a fixed default owner ID.

Figure 6-25 shows the format of the statement.

```
VRSOWNER oooooooo
*        *
1        10
```

Figure 6-25 EDGCDYNM SYSIN: VRSOWNER control statement

In this statement:

oooooooo Is a 1 to 8 character alphanumeric field beginning with an alphabetic letter. A value of AUTH indicates that the TLMS authorization field should be used.

If a VRSOWNER statement is not supplied, the TLMS authorization field is used if valid; otherwise, the default volume owner ID is used.

Only one VRSOWNER statement should be specified; if not, the last statement encountered is used.

Program execution

Example 6-10 shows the sample JCL to execute the extract process for TLMS. See SAMPLIB member EDGJDYNM for the supplied sample JCL.

Archived

Example 6-10 Sample JCL to execute EDGCDYNM

```
//JOBNAME JOB .........
//*
//* The following step assembles and executes the supplied
//* EDGCDYNM assembler code from SAMPLIB. Alternatively
//* you can remove this step and use a load module you have
//* created into common load library.
//*
//* If your system does not use High Level Assembler please
//* change the program name to the name of your assembler.
//*
//DEXTRACT EXEC PGM=EDGCDYNM
//*STEPLIB  DD   DISP=SHR,DSN=rmm.load
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//*
//* TLMSVMF is the TLMS volume master file
//TLMSVMF  DD   DISP=SHR,DSN=tlms.vmf
//*
//* TLMS016 is the retention selection report (Report 016)
//TLMS016  DD   DISP=SHR,DSN=tlms.tlms016.report
//DEXTLIBD DD   DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTLIB.DATA,
//         RECFM=VB,LRECL=700,AVGREC=K,SPACE=(700,(50,20),RLSE)
//DEXTOWN  DD   DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTOWN.DATA,
//         RECFM=VB,LRECL=512,AVGREC=K,SPACE=(512,(1,1),RLSE)
//DEXTRBIN DD   DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTBIN.DATA,
//         RECFM=VB,LRECL=512,AVGREC=K,SPACE=(512,(10,5),RLSE)
//DEXTVRS  DD   DISP=(,CATLG),DSN=STSGWD.EDGCDYNM.DEXTVRS.DATA,
//         RECFM=VB,LRECL=512,AVGREC=K,SPACE=(512,(1,1),RLSE)
//SYSIN    DD   *
OPTIONS OUTSERV=IGNORE VOLCHAIN=PROCESS VOLLOCN=PROCLOC
VRSNAMEP PREFIX=VRSN
VRSMGMTP PREFIX=VDT
TLMSLOCN TLMSID=TL RMMLOC=LIBRARY
TLMSLOCN TLMSID=DC RMMLOC=LIBRARY
TLMSLOCN TLMSID=CL RMMLOC=LIBRARY
TLMSLOCN TLMSID=D2 RMMLOC=LOCAL
TLMSLOCN TLMSID=FS RMMLOC=REMOTE
TLMSLOCN TLMSID=OS RMMLOC=DISTANT
DFRMMLOC RMMLOC=LOCAL RMMTYP=SHELF BINS=001000
DFRMMLOC RMMLOC=DISTANT RMMTYP=SHELF BINS=002000
DFRMMLOC RMMLOC=REMOTE RMMTYP=SHELF BINS=001000/*
```

Archived

6.6.3 EDGCDYNM program limitations

The following are the current program implementation restrictions for EDGCDYNM:

- A maximum of 100 TLMSLOCN statements are allowed.
- A maximum of 100 DFRMMLOC statements are allowed.
- A maximum of 100 SKIPVOL statements are allowed.
- A maximum of 100 VOLMEDIA statements are allowed.

6.6.4 EDGCDYNM program return codes

A return code is set as follows:

0 Normal completion. No errors or warnings detected.

4 One or more warning messages issued. A valid, converted CDS is produced, but the warning messages should be reviewed to ensure that results are as expected. For example, default values may have been used for conversion when not intended.

8 One or more error messages issued. The CDS may contain invalid data. The errors should be corrected and the conversion process rerun.

16 An unrecoverable error has been encountered causing the conversion process to end early. The errors must be corrected and the conversion process rerun.

6.6.5 EDGCDYNM program messages

DXT002 END OF PROGRAM EDGCDYNM. HIGHEST RETURN CODE WAS xx.

where:

xx is the highest return code set by the program.

Severity: Information
Explanation: The EDGCDYNM program has completed execution.
System Action: None
User Response: If a return code greater than 0 is set, the previous messages should be reviewed and any errors corrected before rerunning the program.

DXT010 TLMS DATA EXTRACTION PARAMETERS USED:

Severity: Information
Explanation: The SYSIN control statements are listed following this message for information purposes.
System Action: None
User Response: None

Archived

DXT011 *ERROR* IN PREVIOUS CONTROL STATEMENT - PARAMETERS IGNORED.

Severity: Error
Explanation: An error was encountered in the previous control statement. This could include an unidentified keyword or keyword value, or a value out of range.
System Action: The indicated control statement is ignored, and default values are assumed where appropriate.
User Response: Correct the control statement and rerun the data extraction program.

DXT012 *ERROR* UNIDENTIFIED DFRMM STORAGE LOCATION - XXXXXXXX.

where:

xxxxxxxx is an DFSMSrmm storage location defined in a TLMSLOCN statement.

Severity: Error
Explanation: The identified storage location name was specified on a TLMSLOCN statement, but there is no corresponding DFRMMLOC statement.
System Action: The TLMS location IDs corresponding to the identified DFSMSrmm location will be defaulted to a location of LIBRARY.
User Response: Correct the corresponding TLMSLOCN statements and/or supply appropriate DFRMMLOC statements and rerun the data extraction process.

DXT013 **CRITICAL ERROR UNIDENTIFIED RECORD TYPE=t IN THE DBD FILE**

where:

t is the record type identifier

Severity: Terminal
Explanation: An unidentified record type was encountered in the DBD.
System Action: The process is ended with a return code of 16.
User Response: Correct the TLMSVMF data set records and rerun the data extraction program. EDGCDYNM is designed to run against the actual or image copy of the DBD. Do not use a backup copy taken using the TLMS utility as input.

DXT014 **CRITICAL ERROR VOLUME xxxxxx IS NOT IN ANY DEFINED VOLUME RANGE**

where:

xxxxxx is the volume serial number

Severity: Terminal
Explanation: Specified volume is not in any defined volume range.
System Action: The process is ended with a return code of 16.
User Response: Correct the TLMSVMF control records and rerun the data extraction program. EDGCDYNM is designed to run against the actual or image copy of the DBD. Do not use a backup copy taken using the TLMS utility as input.

DXT020 *WARN* DUPLICATE RMF DSNAME IGNORED - t ddd.....ddd jjjjjjj

where:

t is the type of RMF entry, Full, Partial or Qualified

Archived

ddd.....ddd is the 44-character duplicate data set name
jjjjjjjj is the optional 8-character job name qualifier

Severity: Warning
Explanation: An RMF entry with an identical data set and job name combination to a previous entry was encountered.
System Action: The indicated RMF entry is ignored.
User Response: Review the generated VRS definitions to ensure these produce the same results as the RMF specifications.

DXT022 *ERROR* UNIDENTIFIED TLMS LOCATION=ll RMF=rrr.....rrr

where:

ll is the unidentified TLMS location
rrr.....rrr is the data set name / job name identifying the RMF definition

Severity: Error
Explanation: An RMF entry with an unidentified location identifier was encountered. If mapping of TLMS locations to DFSMSrmm locations is required, the locations must be defined in TLMSLOCN statements.
System Action: The unidentified location ID is translated to an DFSMSrmm location of HOME.
User Response: Review the indicated RMF entry and supply an appropriate TLMSLOCN statement to map the TLMS ID to an DFSMSrmm location. Alternatively, set the location processing option (VOLLOCN on the OPTIONS statement) to PASS and use the EDGCNVT program to map the location identifiers.

DXT024 *WARN* CRASH PROTECTED DATA SET ddd.....ddd DETECTED ON NON-SCRATCH VOLUME vvvvvv. PLEASE CHECK DATA SET NAME.

where:

ddd.....ddd is the truncated 21 character data set name
vvvvvv is the 6 character volume serial number

Severity: Warning
Explanation: A TLMS crash protected data set was detected on a non—scratch volume. It is assumed that crash protection applies only to the first data set on a multi-file and/or multi-volume set.
System Action: The TLMS prefix 'TLMSII-CRASH-PROTECTED' has been removed from the start of the data set name. The data set name may have been truncated by TLMS. If EDGCDYNM recognizes the data set name is not truncated, the volume is marked as recorded by O/C/EOV, else the text 'PLEASE CHECK DATA SET NAME' appears in the message and the volume is not marked as recorded by O/C/EOV.
User Response: If the data set name is not truncated, you can continue with the conversion. If it may be truncated, you must check for the correct data set name and if necessary, correct the data set name in TLMS and rerun EDGCDYNM to correct the data set name, or use the DFSMSrmm CHANGEVOLUME command to correct the data set name once DFSMSrmm is started with the converted CDS.

DXT030 *WARN* INSUFFICIENT BIN NUMBERS SUPPLIED FOR LOCATION -XXXXXXXX. USED=uuuuuu SUPPLIED=ssssss

where:

xxxxxxxx is an DFSMSrmm storage location

uuuuuu is the number of bin numbers allocated to volumes
ssssss the count of bin numbers supplied

Severity: Warning
Explanation: The PROCBIN processing option was specified on the OPTIONS statement to force allocation of bin numbers to volumes in storage locations. Either no or insufficient bin numbers were specified on the DFRMMLOC statement defining the location.
System Action: Bin numbers will be correctly assigned, but no empty bins will be generated.
User Response: Either supply DFRMMLOC statements with appropriate bin numbers or manually add empty bins after conversion using the DFSMSrmm dialog or ADDBIN TSO subcommand. If no empty bins are generated or added, the DFSMSrmm housekeeping process DSTORE may subsequently fail through lack of empty bins.

DXT031 *WARN* NO TLMSLOCN STATEMENTS SUPPLIED - LOCATION PROCESSING REQUESTED.

Severity: Warning
Explanation: You requested, or used by default, the PROCLOC option but did not specify any TLMSLOCN statements.
System Action: A processing option of PASS is forced; no mapping of TLMS storage locations will be performed.
User Response: Either supply TLMSLOCN statements with appropriate mappings or specifically request a processing option of PASS on the OPTIONS statement.

DXT032 *ERROR* UNIDENTIFIED TLMS LOCATION=ll VOLSER=vvvvvv

where:

ll is the unidentified TLMS location
vvvvvv is the VOLSER

Severity: Error
Explanation: A volume entry with an unidentified TLMS location identifier was encountered. If mapping of TLMS locations to DFSMSrmm locations is required, the locations must be defined in TLMSLOCN statements.
System Action: The unidentified location ID is translated to an DFSMSrmm location of SHELF.
User Response: Review the indicated volume entry and supply an appropriate TLMSLOCN statement to map the TLMS ID to a DFSMSrmm location. Alternatively, set the location processing option (VOLLOCN on the OPTIONS statement) to PASS and use the EDGCNVT program to map the location identifiers.

DXT040 *WARN* ZERO VOLUME SEQUENCE NUMBER SET TO ONE - VOLSER=vvvvvv

where:

vvvvvv is the VOLSER

Severity: Warning
Explanation: A non-scratch volume entry with a zero volume sequence number was encountered. This could cause errors in EDGCNVT and/or DFSMSrmm processing.
System Action: The zero volume sequence number is set to 1 to prevent possible later errors.
User Response: Review the indicated volume entry to ensure that it contains valid information.

Archive

DXT092 *ERROR* INSUFFICIENT NUMBER OF ENTRIES IN VRSMTAB.

Severity: Error.
Explanation: The message is issued when more than the specified VRS management values are in use. VRS management values are built during conversion from the special keyword dates detected in the DBD. A VRS management value is built for each unique date detected.
System Action: A VRS management value K record is built but duplicates are not tracked.
User Response: Correct the data extraction program EDGCDYNM: find the line marked as "= = = = = > VRSMTAB SIZE DEFINITION" and increase the number of entries. Then assemble it and re-run the program. If the number of entries will not be increased, the load of the DFSMSrmm CDS will fail because of duplicate K records.

DXT093 *ERROR* INCORRECT CHAIN POINTER(S) ENCOUNTERED. RECORDS SKIPPED = nnnnnnn

where:

nnnnnnn is the number of records skipped due to errors in chain pointers.

Severity: Error
Explanation: A DBD entry with an incorrect TLMS chain pointer has been encountered.
System Action: The volume records pointed to by the incorrect chain pointers are skipped.
User Response: Run the TLMSVCVS utility to correct chain pointers and re-run the data extraction program.

DXT094 RUN TLMSVCVS UTILITY TO CORRECT CHAIN POINTERS

Severity: Information
Explanation: This message will be follows the DXT093 message.
System Action: None
User Response: Run the TLMSVCVS utility to correct chain pointers and re-run the data extraction program.

DXT095 EDGCDYNM INPUT RECORD COUNTS:

Severity: Information
Explanation: This message will be followed by one or more DXT096 messages indicating the number of records read of a particular record type.
System Action: None
User Response: Information

DXT096 NUMBER OF rrrrrrrrrrr RECORDS READ = nnnnnnnn

where:

rrrrrrrr is the type of record
nnnnnnnn is the number of records read

Severity: Information
Explanation: Information
System Action: None
User Response: Information

Archived

DXT097 EDGCDYNM OUTPUT RECORD COUNTS:

Severity: Information
Explanation: This message will be followed by one or more DXT098 messages indicating the number of records output for a particular record type.
System Action: None
User Response: Information

DXT098 NUMBER OF rrrrrrrr RECORDS WRITTEN = nnnnnnnn

where:

rrrrrrrr	is the type of record
nnnnnnnn	is the number of records output

Severity: Information
Explanation: Information
System Action: None
User Response: Information

6.7 Am I ready to continue?

In this chapter we have covered all of the activities that you should have considered before converting from TLMS to DFSMSrmm. After completing the activities in this chapter, you should have the following valid and tested input to load the DFSMSrmm CDS:

- Input to exits
- Correct PARMLIB options
- Housekeeping jobs to use in a DFSMSrmm environment
- Changed reports
- Extract and convert control statements
- Extracted data from TLMS database

Before continuing, we suggest that you ensure that the extraction of data is error free. The output of the extract program may have error return codes. Read the corresponding error messages to determine the actions required to correct the errors. Then rerun the extraction process (see 6.5, “Sample extraction programs” on page 242).

To have a valid DFSMSrmm CDS, you must stop all tape activity from the time you get the data from the TLMS database until you start DFSMSrmm in record-only mode in parallel with TLMS.

Archived

Before producing a valid conversion, you can use a copy of the TLMS database to test the extraction and conversion process, so you do not have to stop the tape activities every time you find that you have to go back and rerun the extract jobs.

Once you have sufficiently tested the environment, you should have correct, tailored input to the extract program. You can now stop tape activities and produce a valid data extraction from the DBD.

If your previous tests were satisfactory, you should obtain the correct input to the EDGCNVT program and proceed to the next step.

Archived

Archived

7

Building the CDS

In this chapter we describe the DFSMSrmm EDGCNVT program and how to use it to build a DFSMSrmm CDS. The input to the EDGCNVT program consists of records that describe the volumes, data sets, and policies you want to convert to DFSMSrmm. In addition, you can control EDGCNVT program processing by using SYSIN control statements, which can edit the input records.

Objectives of this chapter:

- Explain how to use the EDGCNVT conversion program
- Convert extracted data
- Build a ready-to-use DFSMSrmm CDS containing converted data

At the end of this chapter, you will have built a DFSMSrmm CDS ready for use in any of the following environments:

- Testing
- Parallel running
- Cutover to production

© Copyright IBM Corp. 2004. All rights reserved.

7.1 Preparing to run EDGCNVT

Once you have the extracted data from the TLMS database or any other data source, you must use the EDGCNVT program to convert the data to a format usable in the DFSMSrmm CDS.

After running EDGCNVT, optionally run the EDGCEBIN program. The result of running the EDGCNVT program includes:

- Records ready for being loaded into the DFSMSrmm CDS
- DFSMSrmm ADDVRS subcommands that interpret the processed policies
- Control statements for use with the EDGUTIL utility
- Messages that describe processing problems and list statistics

Before running EDGCNVT, you must check your environment for:

- Migration of system-managed tape volumes
- Different pool prefixes
- The best conversion method for VRS policies

7.1.1 System-managed volumes

The EDGCNVT input records do not contain information to indicate whether a location is a system-managed library or a volume is system-managed. At the time you extracted data from TLMS, you may have supplied a location name for every volume belonging to an automated tape library. However, you probably do not have a specific location name for system-managed libraries in TLMS.

You must supply the specific location information by using EDGCNVT control statements. There are three options:

- LOCDEF

  Use this statement to define a location as being system-managed and whether it is an IBM 3494 or IBM 3495 Automated Tape Library (ATL) or a IBM 3495-M10 Manual Tape Library (MTL).

Note: Any customer that has system-managed volumes and using LOCDEF statements for EDGCNVT SYSIN DD should plan to run EDGUTIL with VERIFY(VOLCAT) to ensure that the DFSMSrmm CDS built by conversion is in sync with the TCDB. You can combine the EDGUTIL VERIFY(ALL) with the VOLCAT, by using EDGUTIL with VERIFY(VOLCAT,ALL). See Example 7-5 for more details on how to verify the DFSMSrmm CDS.

Archived

- IF VOLRANGE

  Use this statement to identify the volumes that are system-managed. If you know the range of volumes that are inside the system-managed library or have the system-managed library as a home location, you can assign a specific location name to this range. You only need to do this if the input records do not already have the correct location and home location names.

- IF STORLOC

  If you have a specific location name for the volumes belonging to the library and it is different from the name to be used under DFSMSrmm, you can translate it to the location name that you plan to use in the DFSMSrmm environment.

For an explanation of these statements, go to 7.2, “EDGCNVT and EDGCEBIN processing” on page 266.

Remember that in a system-managed DFSMSrmm environment, the location name you specify for the tape library must match the name of the library defined to SMS.

7.1.2 Pool prefixes

If you plan to move your volumes using media name criteria, you must ensure that you have correctly supplied media names in the extraction process by using the VOLMEDIA for the hard coded values. This is detailed in Chapter 6, “Data extraction” on page 205.

You can optionally use the IF VMEDIA control statement of the EDGCNVT program to translate the media names in the input records to the values you plan to use with DFSMSrmm. For an explanation of these statements, see 7.2, “EDGCNVT and EDGCEBIN processing” on page 266. Ensure that the media names you use match the values specified on the DFSMSrmm PARMLIB LOCDEF and VLPOOL options.

7.1.3 VRS conversion

When you run EDGCNVT specifying the VRSCMDS DD statement, all ADDVRS commands are built from the input K-Records to match the retention and movement criteria. If you are not satisfied with the VRS records built by the combination of extract programs and EDGCNVT program, you can edit the commands. This could be an ideal opportunity to simplify your retention and movement policies to exploit the DFSMSrmm generic data set name mask capabilities.

Archived

If you edit the commands, you can execute the command stream once you have loaded the remaining data created by EDGCNVT into an DFSMSrmm CDS. Remember to remove the VRSLIST file from the EDGJLOAD job so that the VRS records created by EDGCNVT do not end up in the DFSMSrmm CDS.

See 7.8.2, “Adding VRSs” on page 307, and “Reduce VRS policies” on page 383 for additional information on the VRS conversion process.

7.1.4 Movement tracking date

The data set primary VRS subchain start date for all data sets on a volume is set from the volume movement tracking date. The movement tracking date is currently set from the TLMS outcode date (date assigned to a vault). EDGHSKP VRSEL processing uses this date and the volume current location to determine how and if to process EXTRADAYS VRSs on the first VRSEL run after conversion.

7.2 EDGCNVT and EDGCEBIN processing

EDGCNVT is a batch program that takes the records from the input data set, reformats them, and makes them ready for the DFSMSrmm CDS. The input records are usually the records you created by running one of the sample extract processes, but you can construct the records using any technique or tool that you prefer.

As each record is read, it is passed to the EDGCNXIT user exit, if you have supplied one, where it can optionally be updated or skipped. Upon return from the user exit, EDGCNVT compares each record encountered for applicability to the control statements in the SYSIN data set and modifies the records accordingly. It then generates the output records and data sets, which are ready for postprocessing and ultimately entered into the DFSMSrmm CDS.

The exit is not part of EDGCNVT, but you can write it to modify the input records before EDGCNVT begins to operate on them. A sample exit is provided by DFSMSrmm as member EDGCNXIT of SAMPLIB. The macros that map the records passed to EDGCNXIT are described in Appendix B, “Conversion record layouts” on page 613.

The format of the EDGCNVT input records can be found in Appendix B, “Conversion record layouts” on page 613. The SYSIN data set record format can be found in “SYSIN control statements” on page 270.

7.2.1 Running EDGCNVT and EDGCEBIN

Example 7-1 shows the sample JCL for execution of EDGCNVT. The SYSIN file in the JCL is partly created from the running programs described in the previous chapter.

Example 7-1 Sample EDGCNVT execution JCL

```
//CONVERT JOB ....
//*
//STEP1    EXEC PGM=EDGCNVT,REGION=6M
//STEPLIB  DD   DISP=SHR,DSN=RMM.RMM.LINKLIB
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   DISP=(,CATLG),LRECL=80,RECFM=FB,AVGREC=K,
//         SPACE=(80,(1,1),RLSE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.SYSOUT.DATA
//VRSCMDS  DD   DISP=(,CATLG),LRECL=80,RECFM=FB,AVGREC=K,
//         SPACE=(80,(1,1),RLSE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.VRSCMDS.DATA
//*        * ********************************************** *
//*        * following input is created in the DEXTRACT     *
//*        * processing we described in previous chapter    *
//*        * ********************************************** *
//DEXTRCT  DD   DISP=SHR,DSN=RMM.EDGJotms.DEXTRACT.DATA
//         DD   DISP=SHR,DSN=RMM.EDGJotms.DEXTOWN.DATA
//         DD   DISP=SHR,DSN=RMM.EDGJotms.DEXTVRS.DATA
//         DD   DISP=SHR,DSN=RMM.EDGJotms.DEXTVRS.DATA
//LIBLIST  DD   DISP=(,CATLG),LRECL=1000,RECFM=VB,AVGREC=K,
//         SPACE=(1000,(70,10),RLSE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.LIBLIST.DATA
//OWNLIST  DD   DISP=(,CATLG),LRECL=400,RECFM=VB,AVGREC=K,
//         SPACE=(400,(1,1),RLSE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.OWNLIST.DATA
//BINLIST  DD   DISP=(,CATLG),LRECL=256,RECFM=VB,AVGREC=K,
//         SPACE=(256,(20,10),RLSE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.BINLIST.DATA
//VRSLIST  DD   DISP=(,CATLG),LRECL=256,RECFM=VB,AVGREC=K,
//         SPACE=(256,(20,10),RLSE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.VRSLIST.DATA
//SYSIN    DD   *
IF OWNER EQUALS ‘ ‘ THEN DEFOWNR EQUALS ‘USER001’
LOCDEF MTLNAME MTL
LOCDEF ATLNAME ATL
IF VOLRANGE IS 200000 TO 205000 THEN SET LOCATION ATLNAME
IF VMEDIA EQUALS CART3480 THEN VLPOOLNAME EQUALS CARTS
IF VMEDIA EQUALS CART3490 THEN VLPOOLNAME EQUALS CARTS
IF STORLOC EQUALS  LIB THEN RMMSTORE EQUALS SHELF
IF STORLOC EQUALS LIBR THEN RMMSTORE EQUALS SHELF
/*
//*****************************************************************
```

Archived

```
//*
//* SORT BINLIST BY User-Store Name, User-Store Bin Media Name,
//* User-Store Bin Name
//*
//*****************************************************************
//SORT     EXEC PGM=SORT,REGION=4M
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMMRMM..EDGCNVT.BINLIST.DATA
//SORTOUT  DD   DISP=(,PASS),
//         DSN=RMM.EDGCNVT.BINLIST.DATA.SORT,
//         RECFM=VB,LRECL=256,AVGREC=K,UNIT=SYSDA,
//         SPACE=(256,(20,10),RLSE)
//SYSIN    DD   *
 SORT FIELDS=(7,8,CH,A,15,8,CH,A,23,6,CH,A),DYNALLOC
 INCLUDE COND=((5,1,CH,EQ,C'S'),OR,        Include only Bin Records
               (5,1,CH,EQ,C'R'))           (Empty and Assigned)
 OPTION VLSHRT                             Avoid msg ICE218A
/*
//*****************************************************************
//*
//* RUN EDGCEBIN TO FILL THE GAPS BETWEEN USED BINS
//* WITH EMPTY BINS IN THE BINLIST FILE.
//*
//* NOTE: SPECIFY THE PREFIX LENGTH VALUE VIA PARM='PREFLEN(1)'
//* PREFLEN(0) IS DEFAULT
//*****************************************************************
//CEBIN    EXEC PGM=EDGCEBIN,PARM='PREFLEN(2)'
//SORTBIN  DD   DISP=(OLD,DELETE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.BINLIST.DATA.SORT
//BINLIST  DD   DISP=(MOD,KEEP),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.BINLIST.DATA
//SYSPRINT DD   SYSOUT=*
//
```

We describe the input and output files in the sections that follow.

The sample JCL to run the EDGCNVT and EDGCEBIN programs is provided in SAMPLIB member EDGJCNVT.

You should modify the execution parameter values, data set names, and data set attributes, and specify the SYSIN control statements you need before using the sample jobstream.

Archived

7.2.2 EDGCNVT execution parameter

EDGCNVT has a single, optional execution parameter (Table 7-1).

Table 7-1 EDGCNVT execution parameters

Parameter	Description
nnnn	A number of one or more digits that specifies the maximum number of LOCDEF, VOLRANGE or VOLPRE statements to support. The default value is 500 of each type. Specify a value if you want to increase the limit.

7.2.3 EDGCNVT input files

Table 7-2 lists the input files used by EDGCNVT.

Table 7-2 EDGCNVT input files

ddname	Description
DEXTRCT	This file contains the records that are to be converted to DFSMSrmm format. See section DEXTRCT for information about the input records.
SYSIN	The SYSIN control statements describe the processing that you would like EDGCNVT to perform on the records to be converted. See

DEXTRCT

The DEXTRCT file contains the records that are input to EDGCNVT for conversion to DFSMSrmm record format. You can either use the DFSMSrmm-supplied sample extract programs to create the input records or generate them using any tool or method you prefer. This record-based interface provides you with a clearly defined way to convert volume tape resource data into a format supported by DFSMSrmm.

There are five different records that describe input to EDGCNVT. Refer to Appendix B, “Conversion record layouts” on page 613 for details of the records and their contents:

RECORD TYPE **DESCRIPTION**

L Library. An L-Record contains details of a single volume and the first data set on that volume.

D Data set name. A D-Record contains details of a single data set. One D-Record is required for each additional data set on a volume. If you supply a D-Record for a file, all previous files on the same volume must be specified in D-Records that precede the record.

The L and D-Records have the following sequence requirements:

- Library (L) records for a multivolume set of volumes that must appear in ascending volume sequence order, interrupted only by: Data set name (D) records, in ascending file sequence order, after

each L-Record that is created for the volumes from which the data set originates.

o The count of D-Records that follow after a nonscratch L-record must be one less than the count of data sets in the LVDSNCNT field of that L-Record because each L-Record contains first file data set information.

NOTE: D-Records are not supported for scratch volumes.

E Empty rack and bin. The E-Record contains details of a single, empty shelf location. It can describe either a rack number or a bin number.

O Owner. The O-Record contains details of a single owner ID. The owner may own resources that are being converted, but it need not own any resources.

K VRS. The K-Record contains details of a single policy definition.

The EDGCNVT program input records have the following content requirements:

- The intention is that you will use the DFSMSrmm installation-wide exit, EDGUX100, to check the JCL-specified EXPDT value. You should also use that exit to check for a special date of 98000 (other vendor product's “ignore volume”). When the date is found, the exit requests, through the parameter list, that DFSMSrmm ignore the volume. The implication for migration, and therefore for extract processing, is that no volumes with identical VOLSERs can be input to EDGCNVT.
- Data sets that are retained by special keyword dates in TLMS and are to be associated with a special expiration date VRS management value must either:
  - Contain the VRS management value in either the D-Record DSVRSVAL field or the L-Record LDVRSVAL field in the extract program output data set, or
  - Have the VRS management value satisfied through the user exit (see 7.3, “EDGCEBIN processing” on page 295) by updating the LDVRSVAL field in an L-Record or the DSVRSVAL in a D-Record.
- The security level of data sets and volumes should be specified through either the appropriate fields in the L-Records and D-Records or a series of policy and rules statements in SYSIN. The values for security level should be the same as those established through PARMLIB.

SYSIN control statements

EDGCNVT accepts a series of statements from SYSIN in the format shown in Figure 7-1.

```
IF fieldname1 EQUALS fieldname2 THEN fieldnamex EQUALS fieldnamey
                        or                                 or
                    valuexxxxx                         valueyyyyy
```

Figure 7-1 Convert program VRS input record format

EDGCNVT record create date, time, and system identification

EDGCNVT accepts three control statements as shown in Figure 7-2. If these statements are not included, EDGCNVT acquires the values from the operating system at the time of each execution. It uses these values to date, time stamp, and system identify the output records.

```
RMMCRD EQUALS yyyyddd
RMMCRT EQUALS hhmmsst
RMMCRS EQUALS systemid
```

Figure 7-2 EDGCNVT record create date, time, and system identification

Keyword	Description
yyyyddd	The creation date used on all records converted by EDGCNVT.
hhmmsst	Is the creation time used on all records converted by EDGCNVT. The time is specified as: **hh** hours **mm** minutes **ss** seconds and **tt** tenths of second
systemid	This is the system identifier used to identify the system on which a record is created. The value you specify should match the value you specify on the DFSMSrmm PARMLIB OPTION SYSID operand.

VRS release option processing

Use the statement in Figure 7-3 to force or prevent use of VRS release options EXPIRYDATEIGNORE and/or SCRATCHIMMEDIATE.

```
VRS vo choice
```

Figure 7-3 VRSEL release action

Keyword	Description
vo	VRS release option, one of XDI or SI.

Archived

XDI EXPIRYDATEIGNORE, DFSMSrmm ignores the volume expiration date when VRSEL(NEW) is used and no other vital record specification matches to the data set.

SI SCRATCHIMMEDIATE, DFSMSrmm returns the volume to scratch status when the only pending release action for the volume is to return to scratch.

choice ON or OFF.

ON Add specified VRS release option to each created DSN VRS. Override what was set from the previously running conversion tools.

OFF Never add specified VRS release option to any created DSN VRS. Override what was set from the previously running conversion tools. This is the default.

Note: Not specifying VRS XDI and/or VRS SI means use the value set from previous conversion tools.

Assigning a default release action

Use the statement in Figure 7-4 to assign a release action to any volume that does not have a release action specified. This is the only the IF RETACTN statement that is valid.

```
IF RETACTN EQUALS '00' THEN RECACTN EQUALS SCRPOOL
```

Figure 7-4 RETACTN release action

Keyword	Description
00	A hexadecimal value that indicates there are no release actions.
SCRPOOL	Requests that return to scratch release action be assigned to all volumes that have no release action.

Translating location names

Use this statement to translate the name of locations in L-, E-, and K-Records in the DEXTRCT file to values that you plan to use with DFSMSrmm. Only do this if the location names are to be different.

If, for example, TLMS refers to the tape library by the name TLIBRARY, the statement shown in Figure 7-5 will change that name to SHELF, as DFSMSrmm requires. If TLMS uses OFFSITE1, OFFSITE2, or OFFSITE3, the statement shown in Figure 7-5 will change the names to LOCAL, DISTANT, and REMOTE, respectively.

```
IF STORLOC EQUALS TLIBRARY THEN RMMSTORE EQUALS SHELF
                      or                              or
IF STORLOC EQUALS OFFSITE1 THEN RMMSTORE EQUALS LOCAL
                      or                              or
IF STORLOC EQUALS OFFSITE2 THEN RMMSTORE EQUALS DISTANT
                      or                              or
IF STORLOC EQUALS OFFSITE3 THEN RMMSTORE EQUALS REMOTE
                      or                              or
IF STORLOC EQUALS name     THEN RMMSTORE EQUALS location
```

Figure 7-5 IF STORLOC SYSIN statement

Note: STORLOC and RMMSTORE are in the records produced by input to EDGCNVT.

The normal way to use the IF STORLOC statement is together with a LOCDEF statement. The LOCDEF defines the location name and location type as used with DFSMSrmm. If you do not define locations other than SHELF, LOCAL, DISTANT, and REMOTE with LOCDEF, you cannot use the IF STORLOC statement.

Note: Any customer that has System Managed Tape and using LOCDEF statements for EDGCNVT SYSIN DD should plan to run EDGUTIL with VERIFY(VOLCAT) to ensure that the DFSMSrmm CDS built by conversion is in sync with the TCDB. They could combine the EDGUTIL VERIFY(ALL) with the VOLCAT, by using EDGUTIL with VERIFY(VOLCAT,ALL).

If TLMS refers to VLT1, and you want to use the location name VAULT1 with DFSMSrmm, use the statements shown in Figure 7-6.

```
LOCDEF VAULT1 BINS
IF STORLOC EQUALS VLT1     THEN RMMSTORE EQUALS VAULT1
```

Figure 7-6 IF STORLOC example with LOCDEF

If TLMS refers to the in-house disaster storage area for disaster recovery tapes as ONSITE, the statement shown in Figure 7-7 will change that name to LOCAL.

```
IF STORLOC EQUALS ONSITE   THEN RMMSTORE EQUALS LOCAL
```

Figure 7-7 IF STORLOC onsite example

If TLMS refers to the in-town disaster storage area for disaster recovery tapes as OFFSITE, the statement shown in Figure 7-8 will change that name to DISTANT.

```
IF STORLOC EQUALS OFFSITE  THEN RMMSTORE EQUALS DISTANT
```

Figure 7-8 IF STORLOC offsite example

If TLMS refers to the out-of-town disaster storage area for disaster recovery tapes as IRONMTN, the statement shown in Figure 7-9 will change that name to REMOTE.

```
IF STORLOC EQUALS IRONMTN  THEN RMMSTORE EQUALS REMOTE
```

Figure 7-9 IF STORLOC remote example

If your need embedded blanks in the location name, specify the STORLOC value in quotes, and use the statements shown in Figure 7-10 as a sample.

```
LOCDEF VAULT2 BINS
IF STORLOC EQUALS 'VL T2' THEN RMMSTORE EQUALS VAULT2
IF STORLOC EQUALS 'ON SITE' THEN RMMSTORE EQUALS LOCAL
```

Figure 7-10 IF STORLOC with blank example

Specifying security levels

If you would like to establish a security level for data sets that meet a specific naming convention, use the statement shown in Figure 7-11 as a sample.

```
IF DSN EQUALS SYS1.IC.IC.**          THEN SECLEVEL EQUALS 011
```

Figure 7-11 IF DSN SYSIN statement syntax

Note: Only asterisks (*) are supported as filters, and they must be last in the data set name or VOLSER.

The security level number assigned must be a three-digit number and have a value between 0 and 255 as defined by the DFSMSrmm SECLEV option in PARMLIB.

If you would like to establish a security level for volumes that meet a specific naming convention, use the statement shown in Figure 7-12 as a sample.

```
IF VOL EQUALS SY** THEN SECLEVEL EQUALS 009
```

Figure 7-12 IF VOL SYSIN statement syntax

Specifying and translating owner information

If, as the EDGCNVT program is executing, you would like to have owner information added to those volumes and data sets that currently do not have owner information, use the example shown in Figure 7-13.

```
IF OWNER EQUALS ‘       ‘ THEN DEFOWNR EQUALS ‘ library ‘
IF OWNER EQUALS ‘       ‘ THEN DEFOWNS EQUALS ‘ sur name              ‘
IF OWNER EQUALS ‘       ‘ THEN DEFOWNF EQUALS ‘ 1st name               ‘
IF OWNER EQUALS ‘       ‘ THEN DEFOWND EQUALS ‘ dept num                    ‘
IF OWNER EQUALS ‘       ‘ THEN DEFOWNT EQUALS ‘ int tele’
IF OWNER EQUALS ‘       ‘ THEN DEFOWNU EQUALS ‘ userid ‘
IF OWNER EQUALS ‘       ‘ THEN DEFOWNN EQUALS ‘ nodeid ‘
```

Figure 7-13 IF OWNER SYSIN statement syntax for default owner

The keywords must be in the columns indicated.

Keyword	Description
library	1 to 8-character owner ID to be used as a default value
sur name	1 to 20-character surname or family name for the default owner ID
1st name	1 to 20-character first name for the default owner ID
dept num	1 to 30-character department name for the default owner ID.

Note that in Figure 7-13, the space between the beginning and ending apostrophe is less than 30 characters.

int tele	1 to 8-character internal telephone number for the default owner ID
userid	1 to 8-character user ID for the default owner ID
nodeid	1 to 8-character node name for the default owner ID

If, when the EDGCNVT program is executing, you would like to translate one existing owner to a new value, use the example shown in Figure 7-14 with your new values.

```
IF OWNER EQUALS ‘ownerid‘ THEN NEWOWNR EQUALS ‘ newown ‘
IF OWNER EQUALS ‘       ‘ THEN NEWOWNS EQUALS ‘ sur name             ‘
IF OWNER EQUALS ‘       ‘ THEN NEWOWNF EQUALS ‘ 1st name              ‘
IF OWNER EQUALS ‘       ‘ THEN NEWOWND EQUALS ‘ dept num                    ‘
IF OWNER EQUALS ‘       ‘ THEN NEWOWNT EQUALS ‘ int tele’
IF OWNER EQUALS ‘       ‘ THEN NEWOWNU EQUALS ‘ userid ‘
IF OWNER EQUALS ‘       ‘ THEN NEWOWNN EQUALS ‘ nodeid ‘
```

Figure 7-14 IF OWNER SYSIN statement syntax for translating owner

Keyword	Description
ownerid	1 to 8- character owner ID in the input records to be translated
newown	1 to 8-character owner ID to be used instead of owner ID
sur name	1 to 20-character surname or family name for the owner ID
1st name	1 to 20-character first name for the owner ID
dept num	1 to 30-character department name for the owner ID.

Note that in Figure 7-13, the space between the beginning and ending apostrophe is less than 30 characters.

int tele	1 to 8-character internal telephone number for the owner ID
userid	1 to 8-character user ID for the owner ID
nodeid	1 to 8-character node name for the owner ID

Identifying location names and types

Use the LOCDEF statement in Figure 7-15 to define locations other than the DFSMSrmm built-in storage locations (LOCAL, DISTANT, REMOTE, or SHELF) to the conversion program.

```
LOCDEF loc_name type
```

Figure 7-15 LOCDEF statement

Keyword	Description
loc_name	1 to 8-character location name that will be known to DFSMSrmm after conversion. If the location names in the extract records that are input to EDGCNVT do not match these values, use the IF

STORLOC statement to make EDGCNVT perform the translation.

type One of LOAN, BINS, NOBINS, MTL, or ATL:

- **LOAN** Defines the location as a loan location. If the extract records have the location as a storage location, EDGCNVT clears the assigned bin number field and sets the volume location to its home location and marks the volume on loan.
- **BINS** Identifies the location as an installation-defined shelf-managed storage location. The extract records must have set a bin number for the volume.
- **NOBINS** Identifies the location as an installation-defined non-shelf-managed storage location. If the extract records have set a bin number for the volume, it is ignored.
- **ATL** The location is identified as an automated system-managed library.
- **MTL** The location is identified as a manual system-managed library.
- **HOME** Optionally, specify the keyword HOME to indicate that an installation defined storage location can also be a home location.

  When you specify HOME for a storage location, this affects the processing performed for the IF VOLRANGE statement with LOCATION. Any volume in the defined range has its home location set to the name of this storage location.

Up to 500 LOCDEF statements can be defined in SYSIN.

Implementing storage locations as home locations

You can use storage locations as the home locations for volumes to decide how volumes are shelf managed. To implement storage locations as home locations:

1. Specify LOCDEF store BINS HOME in the EDGCNVT SYSIN statements.
2. Specify IF VOLRANGE with LOCATION to assign location names based on volume ranges.
3. Ensure that LOCDEF commands in RMM PARMLIB include TYPE(STORAGE,HOME) only for those storage locations to also be home locations.

Archive

Assigning media type using IF VOLPRE

Use the statement shown in Figure 7-16 to assign a media type to a range of volumes based on the VOLSER prefix and suffix. The media type must be either CST, ECCST, HPCT, EHPCT, or MEDIA5.

```
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeeee THEN SET CST
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeeee THEN SET ECCST
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeeee THEN SET HPCT
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeeee THEN SET EHPCT
IF VOLPRE IS ppppp AND VOLSUF IS ssssss TO eeeeee THEN SET MEDIA5
```

Figure 7-16 VOLPRE SYSIN statement

Keyword	Description
ppppp	VOLSER prefix. Specify from 1 to 5 characters as a prefix.
ssssss	The starting range of a VOLSER suffix. Specify up to 6 characters; the characters must be right aligned, and the total number of prefix and suffix characters must be 6.
eeeeee	The ending range of a VOLSER suffix. Specify up to 6 characters; the characters must be right aligned, and the total number of prefix and suffix characters must be 6.

Up to 500 IF VOLPRE statements can be defined in SYSIN.

Figure 7-17 shows a sample to set the media type to CST for VOLSERs C00001 to C01000 and ECCST for VOLSERs E00001 to E01000.

```
IF VOLPRE IS C     AND VOLSUF IS  00001 to  01000 THEN SET CST
IF VOLPRE IS E     AND VOLSUF IS  00001 TO  01000 THEN SET ECCST
```

Figure 7-17 VOLPRE example

IF VOLRANGE with LOCATION

Use the VOLRANGE with LOCATION statement in Figure 7-18 to assign a location name to a set of volumes based on the VOLSER prefix. The location must be defined by using the LOCDEF statement, or it must be a built-in location name.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET LOCATION loc_name
```

Figure 7-18 VOLRANGE with LOCATION statement

Keyword	Description
fvolsr	The first VOLSER in the range. Specify a 1- to 6-character VOLSER.
lvolsr	The last VOLSER in the range. Specify a 1- to 6-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.
loc_name	The 1- to 8-character location name that will be known to DFSMSrmm after conversion.

Up to 500 IF VOLRANGE with LOCATION statements can be defined in SYSIN.

IF VOLRANGE with PREFIX

Use the IF VOLRANGE with PREFIX statement (see Figure 7-19) to change the rack number that is used for a set of volumes based on the VOLSER. You can change as many characters in the rack number as you want based on the volume VOLSER.

You can still use the significant parts of the VOLSERs to identify volumes, and the rack numbers allow the scratch pool to be identified as well as the VOLSER.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET PREFIX r_pref
```

Figure 7-19 VOLRANGE with PREFIX statement

Keyword	Description
fvolsr	The first VOLSER in the range. Specify a 1- to 6-character VOLSER.
lvolsr	The last VOLSER in the range. Specify a 1- to 6-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.
r_pref	1- to 5-character prefix value that is to be substituted in the rack number passed in the L-Record to EDGCNVT.

Up to 500 IF VOLRANGE with PREFIX statements can be defined in SYSIN.

For example, you have volumes in the 013500–013999 range in a scratch pool, but you also have a scratch pool with a range of 013000-013499. The RMM VLPOOL prefix of 013* does not provide for two separate pools. However, using VOLRANGE with PREFIX, you can change the rack numbers so that the VLPOOL prefixes are different and get RMM to manage two pools. You can use the PREFIX statement as shown in Figure 7-20.

Archived

```
IF VOLRANGE IS 013000 TO 013496 THEN SET PREFIX P
IF VOLRANGE IS 013497 TO 013999 THEN SET PREFIX S
```

Figure 7-20 VOLRANGE with PREFIX statement

You can still use the significant parts of the VOLSERs to identify volumes, and the rack numbers allow the scratch pool to be identified as well as the VOLSER.

IF VOLRANGE with MEDIANAME

Use the VOLRANGE with MEDIANAME statement in Example 7-21 to assign a media name to a set of volumes based on the VOLSER prefix. The media name should be one that matches those you will define on the VLPOOL and LOCDEF DFSMSrmm PARMLIB options.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET MEDIANAME med_name
```

Figure 7-21 VOLRANGE with MEDIANAME statement

Keyword	Description
fvolsr	The first VOLSER in the range. Specify a 1- to 6-character VOLSER.
lvolsr	The last VOLSER in the range. Specify a 1- to 6-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.
media_name	The one- to eight-character media name that will be known to DFSMSrmm after conversion.

Up to 500 VOLRANGE with MEDIANAME statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

IF VOLRANGE with SGNAME

Use the VOLRANGE with SGNAME statement in Figure 7-22 to assign a storage group name to a set of volumes based on the VOLSER prefix.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET SGNAME sg_name
```

Figure 7-22 VOLRANGE with SGNAME statement

Keyword	Description

fvolsr The first VOLSER in the range. Specify a 1- to 6-character VOLSER.

lvolsr The last VOLSER in the range. Specify a 1- to 6-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.

sg_name The 1- to 8-character storage group name that will be known to DFSMSrmm after conversion.

Up to 500 VOLRANGE with SGNAME statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

IF VOLRANGE with TYPE

Use the VOLRANGE with TYPE statement in Figure 7-23 to assign a volume type of either LOGICAL or STACKED to a range of volumes.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET TYPE type
```

Figure 7-23 VOLRANGE with TYPE statement

Keyword	Description
fvolsr	The first VOLSER in the range. Specify a 1 to 6-character VOLSER.
lvolsr	The last VOLSER in the range. Specify a 1 to 6-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.
type	May be either LOGICAL or STACKED

Up to 500 IF VOLRANGE with TYPE statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

IF VOLRANGE with NORACK

Use the VOLRANGE with NORACK statement in Figure 7-24 to prevent having racks assigned to a specific volume range.

```
IF VOLRANGE IS fvolsr TO lvolsr THEN SET NORACK
```

Figure 7-24 VOLRANGE with NORACK statement

Keyword	Description

Archived

fvolsr	The first VOLSER in the range. Specify a 1- to 6-character VOLSER.
lvolsr	The last VOLSER in the range. Specify a 1- to 6-character VOLSER. You can specify the same value as for fvolsr to have a single volume range.

Up to 500 VOLRANGE with NORACK statements can be defined in SYSIN if no execution parameter is specified. If more are required, you can increase the limit using the optional execution parameter.

Editing the volume media name

Use VMEDIA statement to change the media name, which is defined for a volume when the EDGCVNT input records contain a media name that you do not want to use.

For example, you have volumes to which the extract programs have given a media name of CART3480 and TAPE6250, but you want to use CARTS and REEL as the media names instead. You would code the statements as shown in Figure 7-25.

```
   IF VMEDIA EQUALS CART3480 THEN VLPOOLNAME EQUALS CARTS
   IF VMEDIA EQUALS TAPE6250 THEN VLPOOLNAME EQUALS REEL
```

Figure 7-25 VOLRANGE with VLPOOL statement

Figure 7-26 shows the statements that you can code.

```
IF VMEDIA EQUALS TAPE1600 THEN VLPOOLNAME EQUALS media
IF VMEDIA EQUALS TAPE6250 THEN VLPOOLNAME EQUALS media
IF VMEDIA EQUALS CART3480 THEN VLPOOLNAME EQUALS media
IF VMEDIA EQUALS CART3490 THEN VLPOOLNAME EQUALS media
IF VMEDIA EQUALS CART349E THEN VLPOOLNAME EQUALS media
IF VMEDIA EQUALS CART3590 THEN VLPOOLNAME EQUALS media
```

Figure 7-26 VMEDIA statement with VLPOOLNAME

Keyword	Description
media	Media name to be assigned to all volumes with the specified media name. Media names are 1 to 8 characters, and you must ensure that the value specified matches the VLPOOL media names that you specify in the DFSMSrmm PARMLIB.

Archived

Values TAPE1600, TAPE6250, CART3480, CART3490, CART349E and CART3590 are hard coded values. You can specify only one IF VMEDIA statement for each of these hard coded values. If you specify more than one statement, the last value is used.

Assigning a default media name

Use the VMEDIA statement with MEDIANAME in Figure 7-27 to assign a media name to any volume that does not have a media name value specified.

```
IF VMEDIA EQUALS '        ' THEN MEDIANAME EQUALS media
```

Figure 7-27 VMEDIA with MEDIANAME statement

The keywords must be in the columns indicated:

Keyword	Description
media	Media name to be assigned to all volumes that have no media name. Media names are one to eight characters, and you must ensure that the value specified matches the VLPOOL media names that you specify in the DFSMSrmm PARMLIB.

You can only specify one IF VMEDIA with MEDIANAME statement to set a default value. If you specify more than one statement, the last value is used.

Archived

7.2.4 EDGCNVT output files

Table 7-3 lists the output files created by EDGCNVT.

Table 7-3 EDGCNVT output files

ddname	Description
SYSPRINT	The SYSPRINT file is a mandatory file used by the EDGCNVT utility to list all messages issued during processing.
SYSOUT	The SYSIN control statements required by EDGUTIL to create the DFSMSrmm CDS control record are written to this file. They are used by the EDGJVERR sample job.
VRSCMDS	As each input K-Record is processed, it is converted to the equivalent RMM ADDVRS command, and the command is written to this file.
LIBLIST	EDGCNVT writes the DFSMSrmm volume and data set records to this file.
OWNLIST	EDGCNVT writes the DFSMSrmm owner records to this file.
BINLIST	EDGCNVT writes the DFSMSrmm rack and bin number records to this file.
VRSLIST	EDGCNVT writes the DFSMSrmm VRS records to this file.

SYSPRINT

The SYSPRINT file contains the messages generated during EDGCNVT processing. The messages contain statistics, information, and error details. Use the messages to determine whether processing has been successful and to follow up on any nonzero return code.

SYSOUT

At the end of processing EDGCNVT writes the EDGUTIL control statement to the SYSOUT file. The control statement is used during EDGUTIL CREATE processing to initialize the DFSMSrmm CDS control record and set the correct numbers for rack records. Figure 7-28 shows what the file might contain after processing.

```
CONTROL -
   LBINNO(200) LBINFREE(63) -
   RACKNO(100213) RACKFREE(10115) -
   EXTENDEDBIN(YES) STACKEDVOLUME(YES)
```

Figure 7-28 Example of SYSOUT contents

Archive

VRSCMDS

Each time that EDGCNVT successfully processes a K-Record, in addition to creating an DFSMSrmm VRS record, it can optionally create an ADDVRS subcommand that represents the converted policy (Figure 7-29).

```
RMM ADDVRS DSN('P$SECU.SG003*.**')  NOGDG                          +
           CYCLES                                                  +
           COUNT(    3)                                            +
           DELAY( 0)                                               +
           OWNER(AAAAAAAA)                                         +
           DELETE(1999/365)                                        +
           LOCATION(SY      )                                      +
           STORENUM(    2)                                         +
           PRIORITY( 0)                                            +
           NEXT(VRSN0001)

RMM ADDVRS NAME(VRSN0001)                                          +
           OWNER(AAAAAAAA)                                         +
           DELETE(1999/365)                                        +
           LOCATION(CO      )                                      +
           STORENUM(    1)
```

Figure 7-29 Example of VRSCMDS contents

LIBLIST

The EDGCNVT input records together with modifications made by either the EDGCNVT policy and rules statements or the user exit, result in output data sets containing the following data:

From L-Records EDGCNVT produces:

- Volume information for all volumes in the library and all store locations
- Data set name information for the first data set on volumes.

From D-Records, EDGCNVT produces data set name records for each data set, after the first data set, that originates on a volume.

The output records are ready for loading into the DFSMSrmm CDS.

OWNLIST

From O-Records, EDGCNVT produces owner records for each owner that currently owns or could potentially own one or more volumes in the library or storage locations or VRS records.

BINLIST

From E-Records, EDGCNVT produces:

Archived

– Rack information for empty racks that exist in the library
– Bin information for empty bins that exist in any store.

From L-Records, EDGCNVT produces:

– Rack information for scratch and in-use volumes in the library
– Storage location bin information for those volumes that reside in any storage location.

VRSLIST

From the input K-Records, EDGCNVT produces a VRS record containing details of a single retention and movement policy.

7.2.5 EDGCNVT return codes

EDGCNVT sets a return code when it ends. Table 7-4 describes the EDGCNVT return codes.

Table 7-4 EDGCNVT return codes

Return code	Description
0	Normal processing
4	Loan location cannot be converted to a shelf-managed storage location. Bin number 0 is set for a shelf-managed storage location. The number of D-Records supplied for a volume was greater than the number expected.
8	A location name was used that was not defined to the EDGCNVT SYSIN file.
1000	An error occurred that must be corrected before the EDGCNVT program is rerun. See the SYSPRINT file for an explanatory message.

7.2.6 EDGCNVT messages

EDGCNVT-00000 A DEXTRACT Record has been encountered that contains an unidentifiable record type field. The Record appears below and Column 1 is the identifier field. It should be L, O, K, E or D.
<DXT_BASE>

where:

DXT_BASE is the input conversion record

Severity: Error
Explanation: EDGCNVT has detected an invalid conversion record in the DEXTRACT input file.
Action: The job terminates with a return code 1000.
User Response: Re-run the extraction process.

Archived

EDGCNVT-00005 **The number of data sets name records supplied via the DEXTRACT data set greather than anticipated. The data set and volume information about the 'EXTRA" D-record are shown below. VOLSER:** *DSNVOL* **VSEQ:** *VSEQNUM* **LDSNCNT:** *DSCOUNT* **FILE NUMBER:** *ACCOUNT* **DATS SETSEQUENCE:** *FILENUM* **DSN:** *DSNNAME*
DATA SET IGNORED AND CONVERSION CONTINUING.

where:

DSNVOL	is the volume serial number.
VSEQNUM	is the volume sequence number.
DSCOUNT	is the volume records data set count field.
ACCOUNT	is the physical file number.
FILENUM	is the logical file number.
DSNAME	is the data set name.

Severity: Error
Explanation: none
Action: The job terminates with a return code 8.
User Response: Re-run the extraction process.

EDGCNVT-00010 **Home Location Name contained in L-RECORD displayed below does not match any of the Location Names supplied via SYSIN or "SHELF".**
<LVHLOC>
<DXT_BASE>

where:

LVHLOC	is the home location.
DXT_BASE	is the input conversion record.

Severity: Error
Explanation: EDGCVNT is missing a LOCDEF statement in the SYSIN control file to translate the conversion record location name to a location name used in DFSMSrmm.
Action: The job terminates with a return code 1000.
User response: Check the SYSIN control file.

EDGCNVT-00015 **Location specified on a VOLRANGE statement displayed below does not match any of the Location Names supplied via SYSIN or "SHELF".**
<LVHLOC>
<VOLRANGE>
<DXT_BASE>

where:

LVHLOC	is the home location.
VOLRANGE	is the VOLRANGE statement.
DXT_BASE	is the input conversion record.

Severity: Error
Explanation: No matching SYSIN LOCDEF statement has been found to match the location name used in a VOLRANGE statement.
Action: The job terminates with a return code 1000.
User Response: Check the SYSIN control statements for consistency.

Archived

EDGCNVT-00020 **The VMEDIA name supplied via SYSIN is not TAPE1600, TAPE6250, CART3480, CART3490, CART349E or CART3590 and cannot therefore be processed. The name supplied is:**
<MEDNAME>

where:

MEDNAME is the supplied media name.

Severity: Error
Explanation: none
Action: The job terminates with a return code 1000.
User Response: Correct the SYSIN control statement.

EDGCNVT-00025 **The number of storage location names to be converted to Rams LOCAL Disaster Storage Location has exceeded three.**

Severity: Error
Explanation: You are limited to 3 disaster storage area names you can translate to RMM LOCAL location name.
Action: The job terminates with a return code 1000.
User Response: Check your SYSIN control statements.

EDGCNVT-00030 **The number of storage location names to be converted to RMMs DISTANT Disaster Storage Location has exceeded three.**

Severity: Error
Explanation: You are limited to 3 disaster storage area names you can translate to RMM DISTANT location name.
Action: The job terminates with a return code 1000.
User Response: Check your SYSIN control statements.

EDGCNVT-00035 **The number of storage location names to be converted to RMMs REMOTE Disaster Storage Location has exceeded three.**

Severity: Error
Explanation: You are limited to 3 disaster storage area names you can translate to RMM REMOTE location name.
Action: The job terminates with a return code 1000.
User Response: Check your SYSIN control statements.

EDGCNVT-00040 **CONVRT has been passed a data set name that should have been corrected in the analysis of errors in the error data set (ERROR DD) produced by EDGCxLDR or EDGCDYNM. The error DATA SET NAME and VOLUME records are shown below.**
<DSNREC>
<VOLREC>

where:

DSNREC is the data set record
VOLREC is the volume record

Severity: Error

Archived

Explanation: none
Action: The job terminates with a return code 1000.
User Response: Re-run the extraction process.

EDGCNVT-00045 The number of library location names to be converted to RMMs SHELF Library Location has exceeded the limit.

Severity: Error
Explanation: You are limited to 3 disaster storage area names you can translate to SHELF location.
Action: The job terminates with a return code 1000.
User Response: Check and correct your SYSIN control statements.

EDGCNVT-00050 Incorrect IF OWNER statement supplied via SYSIN: *<CNTRL_CARD>*

where:

CNTRL_CARD is the SYSIN control statement

Severity: Error
Explanation: EDGCNVT has detected a syntax error.
Action: The job terminates with a return code 1000.
User Response: Correct the specified SYSIN control statement and re-run the job.

EDGCNVT-00055 Incorrect IF VMEDIA statement supplied via SYSIN: *<CNTRL_CARD>*

where:

CNTRL_CARD is the SYSIN control statement

Severity: Error
Explanation: EDGCNVT has detected a syntax error.
Action: The job terminates with a return code 1000.
User Response: Correct the specified SYSIN control statement and re-run the job.

EDGCNVT-00060 The LOCDEF type supplied via SYSIN is not one of ATL, MTL, BINS or NOBINS and cannot therefore be processed. The name supplied is: *<loc_type>*

where:

loc_type is the location type supplied in the SYSIN control statement.

Severity: Error
Explanation: none
Action: The job terminates with a return code 1000.
User Response: Correct the specified SYSIN control statement and re-run the job.

EDGCNVT-00065 The VRS option supplied via SYSIN is invalid. The VRS option supplied is: *<CNTRL_CARD>*

where:

CNTRL_CARD is the SYSIN control statement

Archived

Severity: Error
Explanation: The VRS release option must be one of XDI or SDI.
Action: The job terminates with a return code 1000.
User Response: Correct the specified SYSIN control statement and re-run the job.

EDGCNVT-00070 **The number of** *<TEXT>* **statements has exceeded** *<TABSIZE>***. Increase the program's capacity by specifying a larger value via the PARM keyword of the JCL EXEC statement.**

where:

TEXT	is one of LOCDEF, VOLRANGE or VOLPRE.
TABSIZE	is the current limit.

Severity: Error
Explanation: EDGCNVT has a single, optional execution parameter to increase the maximum number of LOCDEF, VOLRANGE or VOLPRE statements to support. The default value is 500 of each type.
Action: The job terminates with a return code 1000.
User Response: Specify a value to increase the limit and re-run the job.

EDGCNVT-00075 *<CNTRL_CARD>* **write card in error**
*** VOLRANGE overlap with** *<ENTRY_START>* **to** *<ENTRY_END>*
Correct VOLRANGE input and rerun

where:

CNTRL_CARD	is the SYSIN control statement
ENTRY_START	is the overlapped volume range start volume
ENTRY_END	is the overlapped volume range end volume

Severity: Error
Explanation: At least two VOLRANGE SYSIN control statements were overlapping.
Action: The job terminates with a return code 1000.
User Response: Check and correct the SYSIN control statement.

EDGCNVT-00080 *<CNTRL_CARD>* **write card in error**
*** VOLRANGE is incorrect**
Correct VOLRANGE input and rerun

where:

CNTRL_CARD	is the SYSIN control statement

Severity: Error
Explanation: EDGCNVT has detected an invalid SYSIN control statement where the start volume serial number is greater than the end volume serial number.
Action: The job terminates with a return code 1000.
User Response: Correct the SYSIN control statement and re-run the job.

EDGCNVT-00085 **RMMSTORE value does not match to any location name provided with LOCDEF. Statement in error is:**
<CNTRL_CARD>

where:

CNTRL_CARD is the SYSIN control statement

Severity: Error
Explanation: none
Action: The job terminates with a return code 1000.
User Response: Correct the SYSIN control statement and re-run the job.

EDGCNVT-00090 No Bin number provided in <*LRSBIN*> field for volume assigned to a shelf managed storage location. Volume <*LVOLSER*> assigned from <*LVLOCNAM*> to home location <*LVHLOC*>

where:

LRSBIN is the conversion record bin
LVOLSER is the conversion record volume serial number
LVLOCNAM is the conversion record location name
LVHLOC is the conversion record home location

Severity: Warning
Explanation: none
Action: The job continues and sets a minimum return code of 4.
User Response: Check and, if you correct the conversion input, re-run EDGCNVT.

EDGCNVT-00095 Invalid <*FIELDNAME*> Date <*FIELDVAL*> found in <*RECORDID*> Record <*RECORDKEY*> Invalid date has been replaced with default date 1999365.

where:

FIELDNAME is the conversion record date field name
FIELDVAL is the conversion record date field value
RECORDID is one of Volume, Dataset, Dataset VRS, Name VRS, Volume VRS or VRS
RECORDKEY is the conversion record

Severity: Error
Explanation: The date was either inconsistent or contained invalid non-numerical characters.
Action: The job continues and sets a minimum return code of 8.
User Response: Check and, if you correct the conversion input, re-run EDGCNVT.

EDGCNVT-00100 Incorrect VRS specification found. The VRS is not created: Data set name mask: <*VDSNAME*> Job name mask: <*VRSJOBN*>

where:

VDSNAME is the VRS data set name
VRSJOBN is the VRS job name

Severity: Error
Explanation: EDGCNVT detected invalid characters either in the data set name mask or in the job name mask. In addition to alphanumeric characters you can use %, ¬ or *.
Action: The job continues and sets a minimum return code of 8.
User Response: Correct the conversion input and re-run the job.

EDGCNVT-00105 Current Location Name contained in L-RECORD displayed below does not match any of the Location Names supplied via SYSIN or "SHELF". Location Name set to "SHELF".

Archived

<LVLOCNAM>
<DXT_BASE>

where:

LVLOCNAM	is the location name in the conversion record.
DXT_BASE	is the input conversion record.

Severity: Warning
Explanation: For a location name specified in the conversion input file there does no corresponding LOCDEF statement exist. The location is set to SHELF.
Action: The job continues and sets a minimum return code of 8.
User Response: none

EDGCNVT-00110 The Location Name contained in the K-RECORD displayed below does not match any of the Location Names supplied via SYSIN or "HOME". (SHELF is not a valid next destination name.) (Do not use LOAN location names.) Location Name set to "HOME".

Severity: Warning
Explanation: For a location name specified in the conversion input file there does no corresponding LOCDEF statement exist. The location is set to the home location.
Action: The job continues and sets a minimum return code of 8.
User Response: none

EDGCNVT-00115 The Location Name contained in the E-RECORD displayed below does not match any of the Location Names supplied via SYSIN or "SHELF".

Severity: Warning
Explanation: For a location name specified in the conversion input file there does no corresponding LOCDEF statement exist.
Action: The job continues and sets a minimum return code of 8.
User Response: none

EDGCNVT-00120 The Rack number limit *<MAXBIN>* **for Location** *<LOCNAM>* **has been reached. No further empty bin records will be created.**

where:

MAXBIN	is the rack number limit
LOCNAM	is the location name

Severity: Error
Explanation: none
Action: The job continues and sets a minimum return code of 4.
User Response: none

EDGCNVT-00125 Loan location *<LONLOC>* **cannot be converted to a storage location with BINS for volume** *<VOLSER>*

where:

LONLOC	is the loan location
VOLSER	is the volume serial number

Severity: Warning
Explanation: A loan location cannot be a bin-managed location.
Action: The job continues and sets a minimum return code of 4.
User Response: Check the conversion input and, if you correct the data set name, re-run the job.

EDGCNVT-00130 **Empty bin record** *<ESSTART>* **ignored for location** *<ESLOCNAM>* **type** *<LOCTYPE>* **medianame** *<EVMEDIA>*

where:

ESSTART	is the empty bin record bin
ESLOCNAM	is the empty bin record location
LOCTYPE	is one of ATL, MTL or NOBINS or LOAN
EVMEDIA	is the empty bin media name

Severity: Information
Explanation: You can't have bins in an SMS-managed library, on loan or in a NONBINS location.
Action: The job continues.
User Response: Check the conversion input and, in case, re-run the extraction process.

EDGCNVT-00135 **Bin Number "000000" used in L-RECORD displayed below is invalid for location - home location used** *<LVLOCNAM>*

where:

LVLOCNAM	is the conversion record location name

Severity: Warning
Explanation: none
Action: The job continues and sets a minimum return code of 4.
User Response: none

EDGCNVT-00140 **EDGCNVT PROCESSING STARTED**

Severity: Information
Explanation: none
Action: none
User Response: none

EDGCNVT-00145 **The Number of Data Set Name Records supplied via the DEXTRACT data set is not as expected. It should be Volume Records data set count field (LDSNCNT) minus one. The Volume information is shown below.**
Volser: *<VVOLSER>* **VSEQ:** *<VSEQNUM>* **LDSNCNT:** *<DSCOUNT>*
Actual Count: *<ACCOUNT>*
Highest file number: *<FILENUM>* **PREVVOL:** *<MVPVOL>*
Count corrected and conversion continuing.

where:

VVOLSER	is the volume serial number
VSEQNUM	is the volume sequence number
DSCOUNT	is the volume records data set count field
ACCOUNT	is the physical file number
FILENUM	is the logical file number

Archived

PREVVOL is the previous volume serial number

Severity: Warning
Explanation: none
Action: The job continues and sets a minimum return code of 4.
User Response: none

EDGCNVT-00150 **A data set name containing leading blanks has been detected on a non scratch volume. Data set name has been cleared and data set count has been set to zero. Conversion continues.**
Volser: *<VOLSER>*
Data set name: *<DSNAME>*

where:

VOLSER is the volume serial number
DSNAME is the data set name

Severity: Warning
Explanation: EDGCNVT has detected an invalid data set name for a non-scratch single-file volume.
Action: The job continues and sets a minimum return code of 4.
User Response: Check the conversion input and, if you correct the data set name, re-run the job.

EDGCNVT-00155 **A data set name with leading blanks has been detected on a non scratch volume containing more than one data set. No corrections have been made. Please check and correct the data set name. Conversion continues.**
Volser: *<VOLSER>*
Data set name: *<DSNAME>*

where:

VOLSER is the volume serial number
DSNAME is the data set name

Severity: Error
Explanation: none
Action: The job continues and sets a minimum return code of 8.
User Response: Correct the data set name and re-run the job.

EDGCNVT-00160 **No closing quote for STORLOC value or STORLOC value is more than 8 characters. Statement in error is:**
<CNTRL_CARD>

where:

CNTRL_CARD is the current processed SYSIN control statement

Severity: Error
Explanation: none
Action: The job terminates with a return code 1000.
User Response: Correct the SYSIN control statement and re-run the job.

EDGCNVT-00165 **** WARNING ** EDGCNVT is not APF authorized no mask or TDSI validation performed.**

Severity: Warning

Archived

Explanation: none
Action: The job continues and sets a minimum return code of 16.
User Response: EDGCNVT should reside in an authorized library and be linkedited to AC(1).

7.3 EDGCEBIN processing

EDGCEBIN is a batch program that appends the BINLIST output file to fill the gaps between used bins created by EDGCNVT, with new empty bin records.

Example 7-2 shows the sample JCL for execution of EDGCEBIN.

Example 7-2 Sample EDGCEBIN execution JCL

```
//*****************************************************************
//*     SORT BINLIST BY User-Store Name, User-Store Bin Med        *
//*                  User-Store Bin Name                          *
//*****************************************************************
//SORT     EXEC PGM=SORT,REGION=4M
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMM.EDGCNVT.BINLIST.DATA
//SORTOUT  DD   DISP=(,PASS),
//         DSN=RMM.EDGCNVT.BINLIST.DATA.SORT,
//         RECFM=VB,LRECL=256,AVGREC=K,UNIT=SYSDA,
//         SPACE=(256,(20,10),RLSE)
//SYSIN    DD   *
 SORT FIELDS=(7,8,CH,A,15,8,CH,A,23,6,CH,A),DYNALLOC
 INCLUDE COND=((5,1,CH,EQ,C'S'),OR,        Include only Bin Re
              (5,1,CH,EQ,C'R'))            (Empty and Assigned)
 OPTION VLSHRT                             Avoid msg ICE218A
/*
//*****************************************************************
//*     RUN EDGCEBIN TO FILL THE GAPS BETWEEN USED BINS            *
//*                       WITH EMPTY BINS IN THE BINLIST FILE.   *
//*                                                                *
//*     NOTE: SPECIFY THE PREFIX LENGTH VALUE VIA PARM='PRE        *
//*             PREFLEN(0) IS DEFAULT                              *
//*****************************************************************
//CEBIN    EXEC PGM=EDGCEBIN,PARM='PREFLEN(2)'
//SORTBIN  DD   DISP=(OLD,DELETE),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.BINLIST.DATA.SORT
//BINLIST  DD   DISP=(MOD,KEEP),UNIT=SYSDA,
//         DSN=RMM.EDGCNVT.BINLIST.DATA
//SYSPRINT DD   SYSOUT=*
//
```

Archived

The sample JCL to run the EDGCEBIN program is provided in SAMPLIB member EDGJCNVT.

You should modify the data set names and data set attributes and specify the SYSIN control statements you need before using the sample jobstream.

7.3.1 EDGCEBIN execution parameter

EDGCEBIN has a single, optional execution parameter (Table 7-5).

Table 7-5 EDGCEBIN execution parameters

Parameter	Description
PREFLEN(n)	n is a number of one digit, from 0 to 5, that specifies the prefix length of storage bin numbers in the DFSMSrmm storage location. l = 6 - n, where n is the length of the bin numbers used in the EDGCEBIN program. For example, if you convert from TLMS to DFSMSrmm and want to consider the TLMS storage location id as a prefix and cabinet and box numbers as a bin number, specify PREFLEN(2). The default value is PREFLEN(0).

7.3.2 EDGCEBIN input files

Table 7-6 lists the input files used by EDGCEBIN.

Table 7-6 EDGCEBIN input files

ddname	Description
SORTBIN	This temporary file is created after running EDGCNVT (Table 7-3 on page 284) and contains the bin records from the BINLIST file sorted by User-Store Name, User-Store Bin Media Name, and User-Store Bin Name.

7.3.3 EDGCEBIN output files

Table 7-7 lists the output files used by EDGCEBIN.

Table 7-7 EDGCEBIN output files

ddname	Description
SYSPRINT	The SYSPRINT file is a mandatory file used by the EDGCEBIN program to list all messages issued during processing.
BINLIST	EDGCEBIN appends the DFSMSrmm empty bin number records to this EDGCNVT output file to fill the gabs between used bins with empty bins.

Archived

7.3.4 EDGCEBIN return codes

EDGCEBIN sets a return code when it ends. Table 7-8 describes the EDGCEBIN return codes.

Table 7-8 EDGCEBIN return codes

Return Code	Description
0	Normal processing
4	The SORTBIN file is empty
8	Some records of the SORTBIN file contain non-digital bin numbers, or incorrect execution parameter value specified. see the SYSPRINT file for more information.
12	Syntax error in the execution parameter. See the SYSPRINT file for more information.

7.3.5 6.3.5 EDGCEBIN messages

EDGCEBIN-00000 INCORRECT INPUT PARAMETER: *input_param*

where:

input_param is the EDGCEBIN execution parameter

Severity: Terminal
Explanation: Either the provided execution parameter is not syntactical correct or the prefix length number is not in the defined range.
Action: The step is ended with a return code of 12.
User Response: Correct the execution parameter and restart the job.

EDGCEBIN-00005 DD SORTBIN INPUT FILE IS EMPTY

Severity: Information
Explanation: The supplied SORTBIN input file does not contain any bin records.
Action: The step is ended with a return code of 4.
User Response: Check the DD card. If the BINLIST data set created **<input_param>**with EDGCNVT does not contain any rack or bin records there is no need to run EDGCEBIN.

EDGCEBIN-00010 NON-DIGITAL BIN NUMBER IN THE SORTBIN FILE ENTRY:
<bin_record>
OR INCORRECT INPUT PARAMETER SPECIFIED:
<input_param>

where:

bin_record is the SORTBIN input record
input_param is the EDGCEBIN execution parameter

Severity: Error

Archived

Explanation: Some records of the SORTBIN file contain non-digital bin numbers, or incorrect execution parameter value specified.
Action: EDGCEBIN sets a return code of 8 and continues processing.
User Response: Correct the input and rerun the program.

7.4 EDGRACS processing

If you have no UXTABLE you can skip this section.

EDGRACS is a REXX exec that translates the VRS management values generating ACS statements and placing them into the data set designated by DD name EDGACS, and also generating RMM ADDVRS commands and placing them in the data set designated by DD name EDGCMD. EDGACS contains ACS FILTLIST and WHILE-SET statements. The user can update the statements based on his required policies and merge them with the existing ACS routines. EDGCMD contains RMM ADDVRS commands for RETPD entries from VRSTABLE, which updates the RMM CDS. Before running the commands, the user can manually update the ADDVRS statements according to the required policies.

Example 7-3 shows the sample JCL for execution of EDGCEBIN.

Example 7-3 Sample EDGRACS execution JCL

```
//EDGRACS  EXEC PGM=IKJEFT01,PARM='%EDGRACS',DYNAMNBR=16
//SYSPROC  DD   DISP=SHR,DSN=SYS1.SEDGEXE1
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
//VRSTABLE DD   DISP=SHR,DSN=SYS1.SAMPLIB(EDGCVRSF)
//         DD   DISP=SHR,DSN=RMM.VRSTABLE.DATA
//         DD   DISP=SHR,DSN=SYS1.SAMPLIB(EDGCVRSE)
//EDGACS   DD   DSN=RMM.EDGACS.DATA,DISP=(,CATLG),
//         UNIT=SYSDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
//EDGCMD   DD   DSN=STSGWD.EDGCSRDS.EDGCMD.EXEC,DISP=(MOD,CATLG),
//         UNIT=SYSDA,SPACE=(TRK,(1,1)),LRECL=80,RECFM=FB
```

The sample JCL to run the EDGRACS REXX exec is provided in SAMPLIB member EDGJSRDS.

Archived

7.4.1 EDGRACS input files

Table 7-9 lists the input files used by EDGRACS.

Table 7-9 EDGRACS input files

ddname	Description
VRSTABLE	You VRSTABLE, EDGCVRSF and ESGCVRSE source macros.

7.4.2 EDGRACS output files

Table 7-10 lists the output files used by EDGRACS.

Table 7-10 EDGRACS output files

ddname	Description
EDGACS	All containing ACS statements and filter lists for management class processing. It also contains a note with a recommendation of further EDGUX100 preparation depending on the results of the translation.
EDGCMD	The generated RMM ADDVRS commands to create all needed vital record specifications (VRSs) retention entries. Before running the commands you can manually update the ADDVRS statements according to your required policies.

For detailed information about EDGRACS processing, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

7.4.3 EDGRACS return codes

EDGCEBIN sets a return code when it ends. Table 7-8 describes the EDGCEBIN return codes.

Table 7-11 EDGCEBIN return codes

Return Code	Description
0	Normal processing
4	Warning message has been issued. Probably wrong input data. See joblog
8	Error message has been issued. Wrong input data. See joblog
12	Input or output error when reading or writing a file.

Archived

7.5 EDGCNXIT user exit

EDGCNXIT, a user exit called from EDGCNVT, can be used at your discretion. There are three calls to the user exit in the EDGCNVT program:

- The first call is when the exit is loaded and allows the exit to initialize for processing. For example, it can GETMAIN storage, open a file, and prepare for processing.
- The second call is made each time a record is read from the DEXTRCT file, allowing you to modify the record if necessary. Each record is passed to the exit before EDGCNVT has processed it. Upon return from the exit, processing within EDGCNVT continues, using the updated DEXTRCT record.
- The third call is made at end of processing to allow the exit to issue any messages or free any GETMAINed storage.

The exit is not part of EDGCNVT, but you can write it to modify the input records before EDGCNVT begins to operate on them. A sample exit is provided by DFSMSrmm as member EDGCNXIT of SAMPLIB. The macros that map the records passed to EDGCNXIT are described in Appendix B, “Conversion record layouts” on page 613.

You would use this exit, for example, to add information that TLMS does not provide. A specific case would be the addition of a VRS name (in the LDVRSVAL field of a library record or the DSVRSVAL field of a data set record) to be associated with a tape data set.

Other uses of this exit might be to alter volume ownership by updating the LVOWNER field of L-Records or to alter and/or specify the security level to be associated with a data set by updating the DSNSCLV field of a D-Record and the LVSECLEV field of an L-Record.

Archival

7.5.1 EDGCNXIT input

EDGCNXIT is passed a parameter list. The address of the parameter list is in register1 on entry. The format of the parameter list is:

Offset	Value
+0	Function code. A word containing the function code.
0	Initial call. No record is being passed. Use this exit call to perform any initialization that might be required. The sample exit obtains a work area for use on all exit calls.
4	A record call. The address of a record is passed.
8	Final call. No record is passed. Use this exit call to perform any cleanup necessary. The sample exit frees the work area it obtained.
+4	The return code. This is an output field to be set by the exit.
+8	The address of the record passed to the exit.
+12	A word you can use in the exit for any purpose. The word is set to zero on the initial call and is never changed by EDGCNVT. You can use it to save a work area address or remember any values you want to have from one exit call to the next.

7.5.2 EDGCNXIT output

You can use the EDGCNXIT exit to update the contents of the record passed to you by means of the parameter list.

Set the return code in the parameter list to inform EDGCNVT of the processing you want performed.

Table 7-12 shows the possible return codes from the EDGCNXIT exit execution.

Table 7-12 EDGCNXIT return codes

Return Code	Description
0	Process the record. The exit may or may not have updated the record.
8	Skip this record. If you skip a volume record, remember to skip the data set records that follow it for data sets on the same volume.
16	Invalid parameter list was passed to the exit. This return code causes EDGCNVT processing to end; no more records are processed.

Archive

7.6 DFSMSrmm CDS creation

After the successful execution of the EDGCNVT program, you must run the EDGJLOAD job and create the DFSMSrmm CDS control record to prepare the CDS for parallel running and validation of the conversion process.

7.6.1 Run EDGJLOAD

The SAMPLIB member EDGJLOAD sorts the output records from the EDGCNVT run, sorts the owners generated and removes duplicates, processes the records with the DFSMSrmm EDGCVOVL utility, and finally sorts them using the DFSORT program.

The job allocates the DFSMSrmm CDS and loads the records with an IDCAMS REPRO operation.

Example 7-4 shows sample JCL for EDGJLOAD.

Example 7-4 Sample EDGJLOAD JCL

```
//*JOBNAME JOB .....
//*
//*       $01= ,110,940422,MWW: make the JCL work
//*
//SORT1    EXEC PGM=SORT,REGION=4M
//NEWOWN   DD   DISP=(MOD,DELETE),SPACE=(TRK,1),UNIT=SYSDA,
//         DSN=RMM.EDGCVOVL.INTERM.DATA
//SYSOUT   DD SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMM.EDGCNVT.LIBLIST.DATA
//SORTOUT  DD   DISP=(,PASS),AVGREC=K,SPACE=(1000,(70,10),RLSE),
//         DSN=&&SORTLIB,UNIT=SYSDA
//SYSIN    DD   *
SORT FIELDS=(5,1,CH,D,213,8,CH,A,7,6,CH,A),DYNALLOC
/*
//SORT2    EXEC PGM=SORT,REGION=6M,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMM.EDGCNVT.OWNLIST.DATA
//SORTOUT  DD   DSN=&&SORTOWN,
//         DCB=*.SORTIN,
//         UNIT=SYSDA,DISP=(,PASS),
//         SPACE=(400,(1,1),RLSE),AVGREC=K
//SYSOUT   DD   SYSOUT=*
//SYSIN    DD   *
SORT FIELDS=(6,8,CH,A),DYNALLOC
SUM FIELDS=NONE removes duplicates
END
/*
//CVOVL    EXEC PGM=EDGCVOVL,REGION=4M,COND=(0,NE)
```

Archived

```
//STEPLIB  DD   DISP=SHR,DSN=SYS1.LINKLIB
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//CTLOUT   DD   DISP=(OLD,PASS),DSN=&&SORTLIB
//OWNOUT   DD   DISP=(OLD,PASS),DSN=&&SORTOWN
//NEWOWN   DD   DISP=(,PASS),DSN=RMM.EDGCVOVL.INTERM.DATA,
//         AVGREC=K,SPACE=(1000,(100,30),RLSE),UNIT=SYSDA
/*
//MERGE    EXEC PGM=SORT,REGION=4M,COND=(0,NE)
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMM.EDGCNVT.LIBLIST.DATA,LRECL=2044
//         DD   DISP=SHR,DSN=RMM.EDGCNVT.BINLIST.DATA
//         DD   DISP=SHR,DSN=RMM.EDGCVOVL.INTERM.DATA
//         DD   DISP=SHR,DSN=RMM.EDGCNVT.VRSLIST.DATA
//SORTOUT  DD   DISP=(,PASS),AVGREC=K,SPACE=(1000,(100,30),RLSE),
//         DSN=&&MERGE,UNIT=SYSDA
//SYSIN    DD   *
SORT FIELDS=(5,56,CH,A),DYNALLOC
/*
//DELDEF   EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,VOL=SER=DFRMMA,UNIT=3390
//SYSIN    DD   *
  DELETE RMM.RMM.CDS CLUSTER
  DEFINE CLUSTER (NAME(RMM.PROD.CDS)
                  FILE(MASTER)                  -
                  FREESPACE(15 0)               -
                  KEY(56 0)                     -
                  REUSE                         -
                  RECSZ(512 9216)               -
                  SHR(3 3)                      -
                  VOLUMES(DFRMMA))              -
         DATA    (NAME(RMM.PROD.CDS.DATA)       -
                  CYLINDERS(300 300)            -
                  CISZ(18432)                   -
                  FILE(MASTER))                 -
         INDEX   (NAME(RMM.PROD.CDS.INDEX)      -
                  CYLINDERS(30 30)              -
                  FILE(MASTER))
  SET MAXCC = 0
/*
//LOAD     EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//IN       DD   DISP=(OLD,PASS),DSN=&&MERGE
//OUT      DD   DISP=SHR,DSN=RMM.RMM.CDS,
//         AMP=('BUFND=50,BUFNI=50')
//SYSIN    DD   *
REPRO INFILE(IN) OUTFILE(OUT)
/*
```

Archived

Before running this EDGJLOAD job, you can tailor some parameters, such as the space allocation values for the DFSMSrmm CDS. To choose the space allocation values you can get information from the reports you generated during the environment analysis phase. You need to know how many volumes and data sets you have, how many locations you must define, and the number of VRSs. The output of the extract program can also be useful in getting the information.

After you have the information, you must calculate the amount of space required by using the space calculation rules in the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

While you are loading the DFSMSrmm CDS using the IDCAMS utility, you may see the following message.

```
IDC3314I - RECORDS OUT OF SEQUENCE
```

Figure 7-30 Message IDC3314I

This message may be issued when duplicate records are loaded into the DFSMSrmm CDS. The loading of duplicate records may occur if errors were detected during the extract process. Refer back to the output produced during the extract to see whether any return codes higher than 4 were produced and ensure that all corrective actions are taken before rerunning the programs.

Note also that IDCAMS REPRO continues on for only four errors and then quits. So you may have more errors to correct than those that IDCAMS identifies.

7.6.2 Create the DFSMSrmm CDS control record

The DFSMSrmm CDS control record contains information about the contents of the DFSMSrmm CDS, such as the number of shelf locations in the library and built-in storage locations.

After you have run EDGJLOAD, you must create the DFSMSrmm CDS control record by using the output from the SYSOUT DD statement of the EDGCNVT program. The SYSOUT DD statement contains the SYSIN control statements that the EDGUTIL utility requires to create the DFSMSrmm CDS control record. For more information about EDGUTIL, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

A sample job for creating the DFSMSrmm CDS control record can be found in the SAMPLIB member EDGJVERR (see Example 7-6).

You run the EDGJVERR job to create the DFSMSrmm CDS control record and verify the DFSMSrmm CDS that was built.

Example 7-5 Sample JCL for CDS create and verify

```
//*JOBNAME JOB .....
//CREATE   EXEC PGM=EDGUTIL,PARM='CREATE'
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//MASTER   DD   DISP=OLD,DSN=RMM.RMM.CDS
//SYSIN    DD   DISP=SHR,DSN=RMM.EDGCNVT.SYSOUT.DATA
//* ***************************************************
//*    SYSIN DD *
//*      CONTROL MASTER(Y) -
//*            RACKNO(    3006) RACKFREE(       0) -
//*            EXTENDEDBIN(YES)  STACKEDVOLUME(YES)
```

7.7 Verifying the DFSMSrmm CDS

You must check the VSAM data set to ensure that all information stored in it is in the correct format, and without inconsistencies. Use the EDGUTIL utility with a job like that shown in Example 7-6.

Example 7-6 Sample JCL for CDS verify processing

```
//*JOBNAME JOB .....
//VERIFY   EXEC PGM=EDGUTIL,PARM='VERIFY(ALL)',
//*ERIFY   EXEC PGM=EDGUTIL,PARM='VERIFY(VOLCAT,ALL)',
//         COND=(0,NE),REGION=64M
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//MASTER   DD   DISP=OLD,DSN=RMM.RMM.CDS,
//         AMP=('BUFND=50,BUFNI=50')
//SYSIN    DD   DUMMY
```

Note: Any customer that has System Managed Tape and using LOCDEF statements for EDGCNVT SYSIN DD should plan to run EDGUTIL with VERIFY(VOLCAT) to ensure that the DFSMSrmm CDS built by conversion is in sync with the TCDB. They could combine the EDGUTIL VERIFY(ALL) with the VOLCAT, by using EDGUTIL with VERIFY(VOLCAT,ALL).

You can find sample JCL in the SYS1.SAMPLIB member EDGJVERR.

Refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405 for more information on the parameters that you can use with the EDGUTIL program.

Archived

Review the output of the verification job carefully to see whether there are error messages that will affect the integrity of the DFSMSrmm CDS. If there are errors that do not allow you to proceed to the parallel running phase, you can try to correct the errors and go back to the beginning of the conversion phase, after the extract process, change the necessary EDGCNVT control statements, and restart the conversion steps.

If you cannot correct the errors, it may be necessary to contact an IBM service representative who can help you in the use of the EDGUTIL MEND utility, which may correct the errors.

Note: It should not be necessary to use the MEND utility regularly. Limit its use to fixing DFSMSrmm CDS problems introduced by errors in the conversion process that initially created the DFSMSrmm CDS. Please, ask for IBM assistance before you use the MEND utility.

7.8 Running DFSMSrmm commands

Once you have successfully verified the integrity of the DFSMSrmm CDS, it may be necessary to run some DFSMSrmm commands to:

- Verify and eventually add bins to storage locations
- Add VRSs if the conversion process has not added them.

Note: These commands cannot be run before starting DFSMSrmm. You must consider preparing the command stream that adds VRSs during this phase of the migration in order to be ready for the parallel running stage.

7.8.1 Adding empty bins

If you use shelf-managed storage locations, you might have been able to add empty bins during the extract process for every storage location defined. If you were not able to add bins, you can add them now by using the DFSMSrmm ADDBIN command.

First you must determine how many bins you need for every storage location, and then execute the command:

Example 7-7 ADD empty bins

```
RMM ADDBIN bin_number LOCATION(myloc) COUNT(n_of_bins)
```

In this command:

bin_number is the bin number that you want to add
myloc is the location name where you add bins
n_of_bins is the number of bins you want to add

7.8.2 Adding VRSs

The VRS conversion can be done either manually or through the conversion process. It may be easier to create the VRS definitions manually than to use the conversion programs when:

- There is a small set of VRS definitions to convert
- The VRS was not built properly by the conversion program and does not meet your needs
- You are using the VRSCMDS file from the EDGCNVT program.

If your installation has only a few VRS rules that can be easily defined, you can use the DFSMSrmm ADDVRS command rather than the VRS conversion process.

In addition, if the VRS rules in TLMS environment cannot be easily converted, such as “cycle by days” retention or some particular GDG retention methods, you must consider using the DFSMSrmm ADDVRS command to add them.

The EDGCNVT program shipped with DFSMSrmm gives you an additional opportunity to build VRS definitions by using the output generated in the VRSCMDS DD file. This method can be useful if you want to check VRS definitions before adding them. You can review the definitions in this file and change the VRS commands as necessary before submitting the command stream.

Note: Remember to delete any previous VRS definition from the DFSMSrmm CDS before adding the VRSs from the VRSCMDS file to avoid duplication or error messages.

For more information about using the DFSMSrmm ADDVRS command, refer to “Reduce VRS policies” on page 383.

Archived

Archived

8

Parallel running and validation

Before you change the DFSMSrmm operation mode to PROTECT, you should successfully test all of your production scenarios when DFSMSrmm is running in RECORD-ONLY or WARNING mode. When DFSMSrmm is running in WARNING mode, you can scan your job or system logs for DFSMSrmm messages. When running in parallel, you can compare the vendor product and DFSMSrmm reports for differences.

Objectives of this chapter:

- List and explain all tasks required to achieve the conversion with minimum risk and no loss of data.
- Verify correct conversion.
- Execute DFSMSrmm and the vendor product together to manage the same volumes.
- Identify and explain tasks required to validate that DFSMSrmm and the vendor product are providing the same function.
- Identify when it is time to start cutover to production.

At the end of this chapter you will be ready to deactivate TLMS and run DFSMSrmm in production. You will have validated that DFSMSrmm meets your needs and provides equivalent function.

© Copyright IBM Corp. 2004. All rights reserved.

8.1 Preparing to run in parallel

To prepare for parallel running, we recommend that you test and validate the conversion in a test environment to ensure that you are ready for parallel running in production.

Note: With OS/390 R10 and lower levels, when running DFSMSrmm with TLMS, be sure to start DFSMSrmm before TLMS, or to include the EDGSSSI program name in IEFSSNxx. Both products update LPA to get control during OPEN. The updates must be done in the correct order for processing to work correctly.

8.1.1 Validate PARMLIB members

Check that the following SYS1.PARMLIB members are correctly updated before a start of DFSMSrmm:

IEFSSNxx	Subsystem definition
IKJTSOxx	TSO/E commands and programs
IFAPRDxx	Product enablement policy
EDGRMMxx	DFSMSrmm options
SMFPRMxx (optional)	System management facilities (SMF) parameters
GRSRNLxx (optional)	Global resource serialization resource name lists

Perform this step once for each MVS image. The details about these members are described in the *MVS Initialization and Tuning Reference,* SA22-7592, and for the use of the MVS commands refer to *MVS System Commands Summary,* SA22-7628.

The method for updating the PARMLIB members is described in 3.1.3, “Update and validate PARMLIB members” on page 62.

8.1.2 IEFSSNxx - Subsystem definition

Enabling the DFSMSrmm subsystem interface ensures that the interface will start every time you IPL the system. This step ensures that users cannot use tapes before the DFSMSrmm subsystem starts. To enable DFSMSrmm, change the IEFSSNxx member of SYS1.PARMLIB. Add EDGSSSI, as the DFSMSrmm subsystem initialization program, as shown in Figure 8-1.

Figure 8-1 shows the correct relative position of the DFSMSrmm subsystem, updated to include the initialization program.

Archive

```
SUBSYS SUBNAME(JES2)        /* JES2 PRIMARY SUBSYSTEM START      */
  PRIMARY(YES) START(YES)
SUBSYS SUBNAME(DFRM)        /* Name of the DFSMSrmm subsystem    */
  INITRTN(EDGSSSI)          /* DFSMSrmm initialization routine   */
SUBSYS SUBNAME(AOPA)        /* Netview                           */
```

Figure 8-1 Defining DFSMSrmm subsystem name

8.1.3 EDGRMMxx - DFSMSrmm options

In this section we highlight which updates are important in EDGRMMxx when DFSMSrmm is running in parallel with TLMS.

Update the EDGRMMxx PARMLIB member definition for DFSMSrmm as shown in Example 8-1. Set or update the following OPTION subparameters:

BACKUPPROC(EDGBKUP) Name of the procedure that you want to be started

MAXRETPD(NOLIMIT) The maximum retention period that a user can request for data sets on volumes

MOVEBY(VOLUME) Move volumes that are required by VRS as a set or as individual volumes

OPMODE(W) Running mode of DFSMSrmm

PREACS Enable or avoid DFSMSrmm Pre-ACS processing

RETAINBY(VOLUME) Retains multi-volume sets as a set or as individual volumes

REUSEBIN Control how DFSMSrmm reuses bins when a volume is moving from a bin

SMSACS Control whether DFSMSrmm calls SMS-ACS processing

TPRACF(N) Type of RACF tape support that DFSMSrmm provides

TVEXTPURGE(EXPIRE) How DFSMSrmm processes volumes to be purged by callers of EDGTVEXT or EDGDFHSM

UNCATALOG(N) Type of catalog support

VRSCHANGE(INFO) Specify the action that DFSMSrmm should take during inventory management if you have made changes to VRSs

VRSEL(NEW) Control how DFSMSrmm inventory management vital record processing uses retention and movement information

Archived

Note: The MOVEBY and RETAINBY value definitions are dependent on the definitions in TLMS.

Example 8-1 Sample EDGRMMxx PARMLIB member

```
OPTION    OPMODE(W)                  /* Warning Mode              */ -
          ACCOUNTING(J)              /* Accounting from JOB       */ -
          BACKUPPROC(EDGBKUP)        /* Name of BACKUP-proc       */ -
          BLP(RMM)                   /* DFSMSrmm controls BLP     */ -
          CATRETPD(0012)             /* catalog retention         */ -
          CATSYSID(*)                /* all catalogs shared       */ -
          CDSID(PROD)                /* control data set id       */ -
          COMMANDAUTH(OWNER)         /* type of authorization     */ -
          DATEFORM(J)                /* Date format               */ -
          DISPDDNAME(LOANDD)         /* DISP ctrl  DD card        */ -
          DISPMSGID(EDG4054I)        /* DISP message number       */ -
          DSNAME(RMM.PROD.CDS)       /* CDS data set name         */ -
          IPLDATE(N)                 /* IPL date checking         */ -
          JRNLNAME(RMM.PROD.JRNL)    /* JRNL data set name        */ -
          JOURNALFULL(75)            /* Percentage JRNL full      */ -
          LINECOUNT(054)             /* Lines per page            */ -
          MASTEROVERWRITE(LAST)      /* Overwriting of a vol      */ -
          MAXHOLD(100)               /* Number of I/O oper.       */ -
          MAXRETPD(NOLIMIT)          /* Maximum retention         */ -
          MEDIANAME(3480)            /* spec. how to move vols    */ -
          MOVEBY(VOLUME)             /* spec. how to move vols    */ -
          MSG(M)                     /* case for message txt      */ -
          NOTIFY(Y)                  /* Notify volume owners      */ -
          PDA(OFF)                   /* PDA is disabled           */ -
          PDABLKCT(255)              /* number of blocks          */ -
          PDABLKSZ(31)               /* bloksize in K             */ -
          PDALOG(OFF)                /* PDA output disabled       */ -
          PREACS(YES)                /* enable EDGUX100 ACS proc.*/ -
          RETAINBY(VOLUME)           /* spec. how to retain vols */ -
          RETPD(0005)                /* Default retention         */ -
          REUSEBIN(STARTMOVE)        /* reuse BIN as soon as pos.*/ -
          SCRATCHPROC(EDGXPROC)      /* ATL/MTL procedure         */ -
          SMFAUD(248)                /* SMF audit records         */ -
          SMFSEC(249)                /* SMF security records      */ -
          SMSACS(YES)                /* enable MV ACS processing */ -
/*        SMSTAPE(UPDATE(EXITS,SCRATCH,COMMAND),PURGE(YES))    ATL*/ -
          SYSID(EGZB)                /* Name of the system        */ -
          TPRACF(N)                  /* RACF tape support         */ -
          TVEXTPURGE(EXPIRE)         /* set an expiration date    */ -
          UNCATALOG(N)               /* Catalog support           */ -
          VRSCHANGE(INFO)            /* No additional action      */ -
```

```
         VRSEL(NEW)                 /* New VRS processing       */ -
         VRSJOBNAME(2)              /* DATASETNAME/JOBNAME      */ -
         VRSMIN(0000000100,WARN)    /* Warn if < 100 VRSs       */
/* ******************************************************************* */
MNTMSG   MSGID(IEF233A)      ID(001) VOLUME(016) RACK(016)
MNTMSG   MSGID(IEF233D)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E K')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E R')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E D')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID(IEF455D)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID(IEC501A)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E K')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E D')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E R')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E RD') ID(001) VOLUME(017) RACK(999)
MNTMSG   MSGID('IEC502E RK') ID(001) VOLUME(017) RACK(999)
MNTMSG   MSGID(IAT5110)      ID(001) VOLUME(044) RACK(999)
MNTMSG   MSGID(IAT5210)      ID(001) VOLUME(050) RACK(999)
MNTMSG   MSGID(IAT5410)      ID(001) VOLUME(020) RACK(999)
/* ******************************************************************* */
REJECT   ANYUSE(*)
/* ******************************************************************* */
VLPOOL   PREFIX(SC*)                                                  -
         TYPE(S)                                                      -
         RACF(N)                                                      -
         MEDIANAME(3590)                                              -
         EXPDTCHECK(N)                                                -
         DESCRIPTION('3590 DEFAULT SCRATCH POOL')
VLPOOL   PREFIX(N*)                                                   -
         TYPE(R)                                                      -
         RACF(N)                                                      -
         EXPDTCHECK(0)                                                -
         DESCRIPTION('TEST POOL FOR NORBERT')
VLPOOL   PREFIX(F*)                                                   -
         TYPE(R)                                                      -
         RACF(Y)                                                      -
         EXPDTCHECK(Y)                                                -
         DESCRIPTION('FOREIGN CARTRIDGES')
VLPOOL   PREFIX(PP*)                                                  -
         TYPE(R)                                                      -
         RACF(Y)                                                      -
         EXPDTCHECK(Y)                                                -
         DESCRIPTION('SOFTWARE CARTRIDGES')
/* ******************************************************************* */
LOCDEF   LOCATION(SAFE)                                               -
         TYPE(STORAGE)                                                -
         PRIORITY(2000)                                               -
         MEDIANAME(*)                                                 -
         MANAGEMENTTYPE(BINS)
```

Archived

```
LOCDEF    LOCATION(SANJOSE)                                          -
          TYPE(STORAGE)                                              -
          PRIORITY(2000)                                             -
          MEDIANAME(*)                                               -
          MANAGEMENTTYPE(BINS)
LOCDEF    LOCATION(NEWYORK)                                          -
          TYPE(STORAGE)                                              -
          PRIORITY(2000)                                             -
          MEDIANAME(*)                                               -
          MANAGEMENTTYPE(NOBINS)
LOCDEF    LOCATION(IBMATL1)                                          -
          TYPE(LIBRARY)                                              -
          PRIORITY(4800)
/* ************************************************************** */
SECCLS    NUMBER(010)                                                -
          NAME(UCL)                                                  -
          SMF(N)                                                     -
          MESSAGE(N)                                                 -
          ERASE(N)                                                   -
          DESCIPTION('UNCLASSIFIED')                                 -
          MASK('**')
SECCLS    NUMBER(050)                                                -
          NAME(IUO)                                                  -
          SMF(Y)                                                     -
          MESSAGE(N)                                                 -
          ERASE(N)                                                   -
          DESCIPTION('INTERNAL USE ONLY')                            -
          MASK('**.IUO.**',                                          -
               '**.INTERNAL.**',                                     -
               '**.INT.**')
SECCLS    NUMBER(100)                                                -
          NAME(IC)                                                   -
          SMF(Y)                                                     -
          MESSAGE(Y)                                                 -
          ERASE(Y)                                                   -
          DESCIPTION('EXTERNAL USE ONLY')                            -
          MASK(**.IC.**)
```

Here are the most important options of the OPTION command in the EDGRMMxx PARMLIB member if DFSMSrmm is running in parallel to TLMS:

OPMODE(W)

Specifies the mode of DFSMSrmm:

W **Warning mode**
In addition to performing the actions described for record-only mode, DFSMSrmm validates magnetic tape volumes as you use

Archived

them. If DFSMSrmm discovers errors, it issues error messages instead of rejecting tapes.

BACKUPPROC(EDGBKUP)

Specifies the name of the procedure you want to be started when the journal percentage full threshold is reached.
Specify a valid alphanumeric procedure name from 1 to 8 characters. If no name is specified, then no automatic command is issued.

COMMANDAUTH(OWNER)

Specifies the type of authorization that DFSMSrmm is to check. Specify OWNER when you expect the owners of volume information and data set information to be able to update their own data sets and volumes using RMM TSO subcommands. Specify DSN when you expect changes to volume and data set information to be authorized using the RACF DATASET class and TAPEVOL class.

DATEFORM(J)

Specifies the date format for messages and reports. Specify the date format preferred in your computing center (see Table 8-1).

Table 8-1 Date formats

Value	Language	Format	Example
A	American	mm/dd/yyyy	12/15/2003
E	European	dd/mm/yyyy	15/12/2003
I	ISO	yyyy/mm/dd	2003/12/15
J	Julian	yyyy/ddd	2003/349

DSNAME(RMM.PROD.CDS)

Specifies the name of the DFSMSrmm control data set (CDS). Specify a name up to 44 characters long.

Note: If you do not specify DSNAME, you must specify the data set name in the MASTER DD statement in the DFSMSrmm started procedure. If you specify a name for both DSNAME and MASTER DD, DFSMSrmm ignores the MASTER DD statement.

JRNLNAME(RMM.PROD.JRNL)

Specifies the name of the journal. Specify a name up to 44 characters long.

Note: If you do not specify JRNLNAME, you can specify a name in the JOURNAL DD statement in the DFSMSrmm started procedure. If you do not specify a journal name in either EDGRMMxx or the started procedure, DFSMSrmm does not provide journaling. If you specify a name in both, DFSMSrmm uses the JRNLNAME value and ignores the JOURNAL DD statement.

JOURNALFULL(75)

Specify JOURNALFULL to define a percentage full threshold of the journal data set. When DFSMSrmm detects that the journal has reached this threshold, DFSMSrmm issues message EDG2107E. DFSMSrmm also issues this message at startup if the journal has already reached the threshold specified. If you specify a backup procedure name on the BACKUPPROC operand, the procedure is started automatically. If you specify a value of 0, DFSMSrmm issues no warning on that system. You can specify different threshold values for sharing systems.

Use only a single system to trigger backup using the journal threshold. If you have multiple systems and the journal threshold is reached, each system can start the backup procedure at the same time if you use the same threshold number. To avoid this situation if you have multiple systems, you can specify different thresholds on each system.

For example, specify a threshold of 75 on the main system where DFSMSrmm is active and then specify thresholds of 80 and 85 for your other systems. You can also disable threshold processing on a system by specifying a zero value. Ensure that the system(s) you select will have DFSMSrmm active and have a high chance of processing DFSMSrmm requests. A system that processes no or few DFSMSrmm requests is a bad choice for automatic backup because DFSMSrmm only checks the journal threshold when a request is processed.

MAXHOLD(100)

Specifies the maximum number of activities DFSMSrmm performs to the CDS before the reserve is released and reacquired. Specify MAXHOLD to minimize the impact that long operations, such as RMM TSO ADD subcommands with large count values, or large searches, have on other users.

MAXRETPD(NOLIMIT)

Specifies the maximum retention period that a user can request for data sets on volumes. Specify NOLIMIT or a value between 0 and 9999 days. When a value between 0 and 9999 days is specified, the value is added to the current

date to determine the maximum allowed expiration date. Specify NOLIMIT to use the dates 99365 or 99366 which means to never expire. If the calculated date is 31 December 1999, the expiration date 1 January 2000 is used.

If the DFSMSrmm ISPF dialog or RMM TSO subcommands are used to specify a retention period or expiration date that exceeds the MAXRETPD value, DFSMSrmm fails the subcommand or panel request.

MOVEBY(VOLUME)

Specify the MOVEBY operand to move volumes that are retained by DFSMSrmm vital record specifications as a set of volumes or as individual volumes:

SET Specify this value when you want volumes moved as a set.

PREACS(NO)

Specify the PREACS operand to control whether DFSMSrmm-supplied and EDGUX100 installation exit-supplied values are input to SMS Pre-ACS processing:

NO Specify NO to avoid DFSMSrmm Pre-ACS processing using the DFSMSrmm EDGUX100 installation exit.

RETAINBY(VOLUME)

Use the RETAINBY option to specify whether DFSMSrmm retains multi-volume sets as a set or as individual volumes:

SET When you retain by set, if any volume in a set is retained by a vital record specification, all volumes in the set are retained as vital records.

RETPD(5)

Specifies the default retention period for all new data sets on volumes. Specify a value between 0 and 9999 days. The specified value is added to the current date to determine the expiration date.

Note: The value you specify for the RETPD operand should be the same as the value you use in TLMS.

REUSEBIN(STARTMOVE)

Specifies to control how DFSMSrmm does reuses bins when a volume is moving from a bin.

Archived

STARTMOVE A bin can be reused as soon as a volume starts moving out of a bin. Extended bin support must be enabled before you can use this operand. For more information, see the *DFSMSrmm Implementation and Customization Guide*, SC26-7405, and refer to the chapter titled “Enabling Extended Bin Support” to enable extended bin support.

Note: The use of STARTMOVE is recommended because most vendor tape management systems will reuse bins directly.

SMSACS(YES)

Specify the SMSACS operand to control whether DFSMSrmm calls SMS ACS processing to enable use of storage group and management class values with DFSMSrmm.

YES Specify YES to enable DFSMSrmm calls to the SMS ACS processing to obtain management class and storage group names. If values are returned by the SMS ACS routines, the values are used instead of the DFSMSrmm and EDGUX100 decisions.

SCRATCHPROC(EDGXPROC)

Specifies the name of the procedure DFSMSrmm starts to replenish scratch volumes in an automated tape library dataserver.

SMSTAPE(UPDATE(EXITS,SCRATCH,COMMAND),PURGE(YES))

Use SMSTAPE to specify how DFSMSrmm updates the TCDB and controls system-managed tape processing:

UPDATE Use UPDATE to select the system-managed tape functions DFSMSrmm provides. The UPDATE operand has three subparameters: EXITS, SCRATCH, and COMMAND. You can specify one or more of the subparameters. When DFSMSrmm is running in PROTECT mode, DFSMSrmm ignores the UPDATE operand and performs processing as if you specified EXITS, SCRATCH, and COMMAND. When DFSMSrmm is running in WARNING or RECORD mode, DFSMSrmm does not update TCDB information unless you request the update. You can specify one or more of the values. When you specify a value, DFSMSrmm performs the updates to the TCDB.

Archived

EXITS Specify when you want DFSMSrmm volume status information to override the OAM volume status during entry processing and you want to use the DFSMSrmm VNL exit.

SCRATCH Specify when you want DFSMSrmm to update the volume status in the TCDB during expiration processing when volumes are returned to scratch status.

COMMAND Specify when you want to use the RMM TSO subcommands or the DFSMSrmm API to update the TCDB.This controls change of status, TDSI and owner information, eject processing and manual cartridge entry processing.

PURGE Use PURGE to control how DFSMSrmm affects the TCDB volume records during EJECT processing. The default is PURGE(ASIS) in all operating modes except MANUAL mode. In manual mode, DFSMSrmm provides no support for eject processing.

YES Specify when you want DFSMSrmm to force the TCDB volume records to be purged at eject time.

TPRACF(N)

Do not create RACF tape profiles.

Attention: DFSMSrmm will manipulate your RACF TAPEVOL and TAPEDSN profiles in each operating mode if the TPRACF function is set to A (automatic) or P (predefined).

TVEXTPURGE(EXPIRE)

Specifies how DFSMSrmm processes volumes to be purged by callers of EDGTVEXT or EDGDFHSM.

EXPIRE Use the EXPIRE option to set the volume expiration date to the current date for volumes to be purged. Use this operand in combination with vital record specifications that use the UNTILEXPIRED retention type. You can then optionally extend retention using the extra days retention type.

Recommendation: When you cut over to production with DFSMSrmm take the default for this option, TVEXTPURGE(RELEASE).

Archive

UNCATALOG(N)

Do not use DFSMSrmm uncatalog processing. If UNCATALOG(N) is set, DFSMSrmm does not uncatalog data sets under any circumstances.

Attention: DFSMSrmm will uncatalog data sets in each operating mode if the UNCATALOG function is set to Y or S.

VRSCHANGE(INFO)

Use VRSCHANGE to specify the action DFSMSrmm should take during inventory management if you have made any changes to vital record specifications. If INFO is set, no action is taken during inventory management. With VERIFY, any change in vital record specifications must be verified by running EDGHSKP with the VERIFY parameter.

INFO No additional processing or actions are required when vital record specification changes occur.

VRSJOBNAME(2)

Specify this operand to select whether DFSMSrmm uses the job name from VRS as the primary or secondary match.

VRSJOBNAME(2) The data set name takes precedence over the job name during VRS match processing. This is the option that most closely matches the use of job names in TLMS.

VRSEL(NEW)

Specify this operand to select the VRS processing method used by DFSMSrmm.

NEW Specify VRSEL(NEW) to exploit the new VRS features such as mixed retention type VRS chains, and VRS release options. We recommend the use of this option because the conversion programs build new types of VRSs.

Specify if you want:

- DFSMSrmm to process retention information in NAME VRSs, release options defined in VRSs, and VRSs chained using the RMM ADDVRS operand.
- If you are converting using the conversion tools and require functions close to that given by TLMS.

For information about the different options for each subparameter, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

Archive

8.1.4 Check the DFSMSrmm procedure

Check the procedure in SYS1.PROCLIB to start the DFSMSrmm subsystem address space. Example 8-2 contains sample JCL.

Example 8-2 Sample DFSMSrmm starting procedure

```
//DFRMM    PROC M=00,OPT=MAIN
//IEFPROC  EXEC PGM=EDG&OPT.,PARM='&M',TIME=1440,REGION=4096K
//PARMLIB  DD   DDNAME=IEFRDER
//IEFRDER  DD   DISP=SHR,DSN=SYS1.PARMLIB
//*MASTER   DD   DISP=SHR,DSN=RMM.PROD.DSET
//*JOURNAL  DD   DISP=SHR,DSN=RMM.PROD.DSET
//EDGPDOX  DD   DISP=SHR,DSN=RMM.&SYSNAME..RMMPDOX
//EDGPDOY  DD   DISP=SHR,DSN=RMM.&SYSNAME..RMMPDOY
```

For an explanation of each sentence in the procedure, refer to 3.1.4, “Creating a starting procedure” on page 85.

In the sample procedure in Example 8-2, the MASTER and JOURNAL DD statements are commented by intention. This is recommended, so you can disable the journal data set during special runs of inventory management. In 8.3.1, “First run of EDGHSKP with PARM(VRSEL)” on page 346, a scenario of disabling the journal data set is described.

8.1.5 Exits

DFSMSrmm provides programming interfaces and source code for installation-wide exits installed as part of DFSMSrmm during the SMP/E apply process.

Since DFSMS/MVS OS/390 V1.4, DFSMShsm and DFSMSrmm communicate directly without calling the ARCTVEXT exit.

If you will share one or more exits between TLMS and DFSMSrmm, you must check if updates are needed in any of the exits to achieve coexistence when running in parallel.

Refer to 3.1.2, “Update installation-wide exits” on page 61 for more information.

8.1.6 CA-Disk interface

You can modify your CA-Disk (previously SAMS:Disk) to support more than one tape management system. To do this, you have to update the CA-Disk SYSPARMs in the following way:

Archived

TAPCTLEXADSTH016 Adding this in the CA-Disk SYSPARM will activate the CA-Disk DFSMSrmm support.

TMSCTLEXADSTH014 Normally this parameter is already defined to support your existing TLMS system.

Note: After the cutover you will change the TMSCTLEX from ADSTH014 to ADSTH016 value, and delete the TAPCTLEX entry from your CA-Disk SYSPARM.

For more information read the CA-Disk product publications.

8.1.7 Verify authorization and security

To protect DFSMSrmm functions, you need to use an external security product, such as RACF. If RACF is not installed, you must provide equivalent function through the SAF interface.

When running on a system with RACF installed, check that DFSMSrmm is defined in the started procedures name table ICHRIN03 or in the RACF STARTED class.

Each DFSMSrmm resource has an entity name prefixed with STGADMIN.EDG. You must define the RACF profiles for these resources in the FACILITY class and grant permission to all users who have a need for the function.

For more information about protecting DFSMSrmm resources, refer to 3.1.6, “Protecting DFSMSrmm resources” on page 89.

Note: Grant the started task procedure (DFRMM) RACF user ID ALTER access authority to STGADMIN.EDG.RESET.SSI so you can disable the DFSMSrmm subsystem by using the command shown in Figure 8-2.

```
S DFRMM,OPT=RESET

P DFRMM
```

Figure 8-2 Disabling the DFSMSrmm subsystem interface

For more information on authorization and security, see the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

8.1.8 Prepare inventory management using EDGHSKP

DFSMSrmm provides the EDGHSKP utility to help you perform inventory management.

We recommend that you use an existing job scheduling product to run the utilities, such as IBM Tivoli Workload Scheduler. Sample Tivoli Workload Scheduler jobs are provided in SYS1.SAMPLIB for running inventory management activities. Refer to 11.1.1, “IBM Tivoli Workload Scheduler overview” on page 434, for more information about how to schedule inventory management tasks.

While DFSMSrmm is running in parallel, schedule the following DFSMSrmm functions:

- Vital record processing (VRSEL)
- Expiration processing (EXPROC)
- Backing up the DFSMSrmm CDS (BACKUP)
- Backing up and clearing the DFSMSrmm journal
- Creating an extract data set (RPTEXT)
- Storage location management (DSTORE)

All of the above functions should run with the same frequency and time as the equivalent functions in your TLMS running in parallel.

Note: You must run the EDGHSKP BACKUP function, as it clears the DFSMSrmm journal data set when the backup of the DFSMSrmm CDS is successful.

In order to keep a record of the EDGHSKP utility processing you can copy the output files such as the MESSAGE file to SYSOUT. Copying the output files to SYSOUT enables you to view the output along with the job and prevents data loss due to multiple runs of EDGHSKP.

EDGHSKP program

Example 8-3 contains sample JCL for the EDGHSKP program.

Example 8-3 Sample EDGHSKP JCL

```
//HSKP     EXEC PGM=EDGHSKP,
//          PARM='VRSEL,EXPROC,DSTORE,BACKUP(DSS),RPTEXT,DATEFORM(E)'
//ACTIVITY DD   DISP=SHR,DSN=RMM.HSKP.ACTIVITY
//MESSAGE  DD   DISP=SHR,DSN=RMM.HSKP.MESSAGES
//REPORT   DD   DISP=SHR,DSN=RMM.HSKP.REPORT.FILE.NAME
//*REPTEXT  DD   DISP=SHR,DSN=RMM.HSKP.EXTRACT.FILE.NAME
//XREPTEXT  DD   DISP=SHR,DSN=RMM.HSKP.XEXTRACT.FILE.NAME
```

```
//SYSPRINT DD   SYSOUT=A
//BACKUP   DD   DISP=(,CATLG),DSN=RMM.PROD.CDSBKUP,UNIT=SYSDA
//JRNLBKUP DD   DISP=(,CATLG),DSN=RMM.PROD.JCNLBKUP,UNIT=SYSDA
```

The sample jobs provided for scheduling inventory management tasks using Tivoli OPC use the EDGHSKP program for performing the required activities.

EDGHSKP input files

Table 8-2 lists the output files used by EDGHSKP.

Table 8-2 EDGHSKP input files

ddname	Description
DSSOPT	Contains DUMP or RESTORE command options used by DFSMSdss during backup processing.

Example 8-4 shows you how to ensure that DFSMSdss does not compress the data, and that the tape hardware is used to compress the records. This example uses EDGHSKP to back up to tape. To use EDGBKUP, the backup data sets must be allocated on DASD.

Example 8-4 Using DSSOPT DD statement

```
//EDGBKUP  EXEC PGM=EDGHSKP,PARM='BACKUP(DSS)'
//SYSPRINT DD   SYSOUT=*
//BACKUP   DD   DISP=(,CATLG),UNIT=TAPE,DSN=RMM.PROD.CDSBKUP(+1),
//         LABEL=(,SL)
//JRNLBKUP DD   DISP=(,CATLG),UNIT=TAPE,DSN=RMM.PROD.JRNLBKUP(+1),
//         LABEL=(2,SL),VOL=REF=*.BACKUP
//DSSOPT   DD   *
CONCURRENT OPTIMIZE(4) VALIDATE
/*
```

EDGHSKP output files

Table 8-3 lists the output files used by EDGHSKP.

Table 8-3 EDGHSKP output files

ddname	Description
ACTIVITY	Contains detailed information about data set related changes DFSMSrmm makes to the control data set during inventory management. This data set is required when you specify the VERIFY parameter. It must be preallocated.

Archived

ddname	Description
BACKUP	Contains the backup copy of the DFSMSrmm control data set. Specify this data set to run backup processing for the control data set. You can back up directly to tape when you specify the BACKUP(DSS) parameter even when DFSMSdss concurrent copy is not available. This data set is required when you specify the BACKUP parameter. You do not need to preallocate this data set.
JRNLBKUP	Contains the backup copy of the DFSMSrmm journal. Specify this data set to run backup processing for the journal. DFSMSrmm uses IDCAMS to back up the journal when you specify the BACKUP(AMS) or BACKUP(DSS) parameter. You can back up directly to tape when you specify the BACKUP(DSS) parameter even when DFSMSdss concurrent copy is not available.and you will backup the JOURNAL. This data set is required when you specify the BACKUP parameter and you will create a backup copy of your journal too. You do not need to preallocate this data set.
MESSAGE	Lists the messages the DFSMSrmm subsystem issues during inventory management. This data set is required. It must be preallocated.
REPORT	Contains a detailed report of DFSMSrmm vital record specification processing. Specify if you want a report when you have specified the VRSEL parameter. It must be preallocated.
REPTEXT	Contains the extract copy of the DFSMSrmm control data set. The extract copy is called the extract data set. You must specify either the REPTEXT DD statement or the XREPTEXT DD statement when you use the EDGHSKP RPTEXT parameter. It must be preallocated.
SYSPRINT	Contains the utility program messages that IDCAMS and ADRDSSU issue when backing up the DFSMSrmm control data set. The SYSPRINT data set is required when you specify the BACKUP parameter. This data set can be a SYSOUT file. You do not need to preallocate this data set.
XREPTEXT	Contains the extract copy of the DFSMSrmm control data set. The extract copy is called the extract data set. You must specify either the REPTEXT DD statement or the XREPTEXT DD statement when you use the EDGHSKP RPTEXT parameter. It must be preallocated.

Archived

Recommendation: Always run DFSMSrmm with the optional files ACTIVITY and REPORT. Maintain these files, as well as the MESSAGE, BACKUP, JRNLBKUP, SYSPRINT, REPTEXT, and XREPTEXT files, as GDGs or copy them to GDGs. You might find that multiple versions of the files are helpful for use in diagnosing problems or for creating reports.

The first step should be the definition of the necessary data sets. The following list shows the different data sets. See 3.4.8, “Allocate inventory management data sets” on page 122 for information on allocating these files.

Before you can use the EDGHSKP utility to run inventory management functions, you must pre-allocate all EDGHSKP output files, except BACKUP and JRNLBKUP. Example 8-5 shows you sample of JCL for EDGHSKP.

Example 8-5 Example of JCL for EDGHSKP.

```
//HSKP     EXEC PGM=EDGHSKP,
//         PARM='VRSEL,EXPROC,DSTORE,BACKUP(DSS),RPTEXT,DATEFORM(E)'
//SYSPRINT DD   SYSOUT=A
//DSSOPT   DD   *
CONCURRENT OPTIMIZE(1) VALIDATE
/*
//BACKUP   DD   DSN=RMM.PROD.CDSBKUP,DISP=(NEW,CATLG),
//         UNIT=SYSDA,AVGREC=U,SPACE=(4096,(1000,500)),
//         LRECL=9000,BLKSIZE=0,RECFM=U
//JRNLBKUP DD   DSN=RMM.PROD.JRNLBKUP,DISP=(NEW,CATLG),
//         UNIT=SYSDA,AVGREC=U,SPACE=(4096,(1000,500)),
//         LRECL=9000,BLKSIZE=0,RECFM=VB
//MESSAGE  DD  DSN=RMM.PROD.MSGS,DISP=SHR
//REPORT   DD  DSN=RMM.HSKP.REPORT.FILE.NAME,DISP=SHR
//ACTIVITY DD  DSN=RMM.HSKP.ACTIVITY.FILE.NAME,DISP=SHR
//*REPTEXT  DD  DSN=RMM.HSKP.EXTRACT.FILE.NAME,DISP=SHR
//XREPTEXT DD  DSN=RMM.HSKP.XEXTRACT.FILE.NAME,DISP=SHR
```

The data set referred to in the BACKUP DD contains the backup copy of the DFSMSrmm CDS. You can back up directly to tape when you specify the BACKUP(DSS) parameter. This data set is required when you specify the BACKUP parameter. You do not need to preallocate this data set.

EXEC parameters in EDGHSKP

Table 8-4 shows the possible EXEC parameters for the EDGHSKP program.

Note: All of the parameters should not be specified during the parallel-run phase of the conversion. Some of the parameters also depend on the setting of the related OPTION parameter in the PARMLIB member.

Table 8-4 EDGHSKP execution parameters

ddname	Description
BACKUP	Backing up the DFSMSrmm control data set and DFSMSrmm journal
CATSYNCH	Update the DFSMSrmm control data set with information from available user catalogs.
DATE	Specify DATE to set the date used for VERIFY processing.
DATEFORM	Set the date format for records that are written.
DSTORE	Performing storage location management
EXPROC	Performing expiration processing
RPTEXT	Creating an extract data set
VERIFY	Request that DFSMSrmm performs a trial run of selected processing.
VRSEL	Processing vital records

BACKUP

Determine the backup technique to use:

DSS Use when you want DFSMSdss to perform the backup. Using DFSMSdss concurrent copy permits the update of the CDS during backup processing so all tape activity can continue during CDS backup processing.

To use DFSMSdss concurrent copy, you must have a concurrent copy environment set up. SnapShot capable devices are also supported by DFSMSdss.

The DFSMSdss operands are specified in the DSSOPT DD statement, in the EDGHSKP and EDGBKUP utilities.

AMS This is the default. DFSMSrmm uses access method services REPRO command to perform backup processing.

CATSYNCH

Specify CATSYNCH to request that DFSMSrmm updates the DFSMSrmm CDS with information from available user catalogs. Before you can synchronize the DFSMSrmm CDS with user catalogs, you must first define system IDs using the EDGRMMxx PARMLIB OPTION CATSYSID operand. Refer to the chapter 8.1.10, “Catalog synchronization” on page 329, or the manual *DFSMSrmm Implementation and Customization Guide*, SC26-7405 for more details.

Archived

DATE

Use this parameter to set the date for VERIFY processing. The current date is the default.

DATEFORM

Use this parameter to override the date format in reports produced in inventory management. The default date format for all date fields is the value specified in the PARMLIB member EDGRMMxx. The value is initially set to J for Julian.

DSTORE

Specify this parameter for storage location management.

EXPROC

Specify this parameter for expiration processing.

RPTEXT

Specify this parameter to create records in extract data set. If you code XREPTEXT DD, then extended extract records are created in the extract data set.

VERIFY

Specify this parameter to request a trial run of vital record processing. DFSMSrmm performs vital record processing, but does not update the control data set on this run of inventory management. You can use this operand to test new vital record specifications that you define to DFSMSrmm. The use of VERIFY in the EDGHSKP utility depends on the setting of the VRSCHANGE parameter in the PARMLIB member EDGRMMxx. The default value of VRSCHANGE is VERIFY.

Note: The VRSCHANGE parameter in PARMLIB member EDGRMMxx should be set to INFO during parallel processing.

VRSEL

Specify this parameter for vital record processing.

Archived

8.1.9 Inventory and movement reporting using EDGRPTD

DFSMSrmm provides inventory and movement reporting using the EDGRPTD report utility. Example 8-6 shows sample JCL to produce reports using the EDGRPTD utility.

Example 8-6 Sample JCL for inventory and movement reporting

```
//RPTD     EXEC PGM=EDGRPTD,
//          PARM='SEC''(INTERNAL USE ONLY)'',DATEFORM(I),LINECOUNT(54)'
//REPTEXT  DD   DISP=SHR,DSN=HSKP.XREPTEXT
//SYSPRINT DD   SYSOUT=*
//INSTVOL  DD   DISP=SHR,DSN=HSKP.INSTVOL
//INSTBIN  DD   DISP=SHR,DSN=HSKP.INSTBIN
//INSTOWN  DD   DISP=SHR,DSN=HSKP.INSTOWN
//TOSTRCK  DD   DISP=SHR,DSN=HSKP.TOSTRCK
//TOSTOWN  DD   DISP=SHR,DSN=HSKP.TOSTOWN
//FMSTBIN  DD   DISP=SHR,DSN=HSKP.FMSTBIN
//FMSTOWN  DD   DISP=SHR,DSN=HSKP.FMSTOWN
//RDYTOSCR DD   DISP=SHR,DSN=HSKP.RDYTOSCR
//SYSOUT   DD   SYSOUT=*
//SCRDATE  DD   DSN=rmm.last.run.date,DISP=OLD
//SCRLIST  DD   DISP=(,CATLG),DSN=rmm.scratch.list,
//          UNIT=SYSDA,SPACE=(TRK,(10,10)),LRECL=121,RECFM=VBA
//NEWSCR   DD   DISP=(,CATLG),DSN=rmm.new.scratch.list,
//          UNIT=SYSDA,SPACE=(TRK,(10,10)),LRECL=121,RECFM=VBA
```

The input file in the EDGRPTD run is the REPTEXT DD, or XREPTEXT DD file created in the EDGHSKP inventory management run with PARM(RPTEXT) specified.

Refer to Chapter 12, “Reporting on your DFSMSrmm environment” on page 485, for more information about the different reports.

8.1.10 Catalog synchronization

Since DFSMS/MVS V1.5 Vital Record Selection (VRSEL) is enhanced by removing the need to perform catalog locates for all data sets that match to a WHILECATALOG VRS. The catalog status of data sets recorded by DFSMSrmm is now maintained dynamically during catalog and uncatalog operations.

This new function cannot be exploited until you enable it by synchronizing the DFSMSrmm control data set (CDS) with your user catalogs using EDGHSKP with PARM=CATSYNCH, and for unshared catalogs, enabling with EDGUTIL.

Archived

The following enhancements are included:

- How you enable catalog synchronization depends on your catalog environment. For fully shared catalogs, you can specify the CATSYSID(*) PARMLIB option in member EDGRMMxx, and run EDGHSKP with PARM='CATSYNCH'. For unshared catalogs, you must specify CATSYSID(*list of sysids*) and run EDGHSKP with CATSYNCH once per group of systems, and then enable synchronization using EDGUTIL with PARM='UPDATE'. The running of EDGHSKP with CATSYNCH is used to synchronize the DFSMSrmm control data set (CDS) with the available ICF catalogs. For full details, refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.
- Once the function is enabled, the VRS WHILECATALOG retention type and the PARMLIB UNCATALOG option no longer rely on locate processing, but use the DFSMSrmm recorded catalog status. If the catalog status of a data set is unknown, VRSEL processing applies the CATRETPD retention period. Only after applying the CATRETPD retention period does DFSMSrmm treat the data set as non-cataloged. A locate is only issued during EDGHSKP CATSYNCH processing.
- The LISTDATASET output and the report extract file data set record contain the catalog status as recorded by DFSMSrmm.
- For environments with unshared catalogs, you can run EDGHSKP CATSYNCH on a subset of catalogs by specifying the CATSYSID PARMLIB OPTION in the EDGRMMxx member.
- For unshared catalog environments EDGHSKP EXPROC return to scratch processing can now be run on a subset of volumes identified by the CATSYSID PARMLIB option.
- LISTCONTROL CNTL can list the date and time the synchronization was last performed. The control dialog is also updated to show date and time.

> **Tip:** We recommend that you enable this function to improve the performance of the daily vital record and expiration processing.

Refer to the manual *DFSMSrmm Implementation and Customization Guide*, SC26-7405, and the chapter titled "Maintaining the Control Data Set" for more details.

Running DFSMSrmm catalog synchronization

Before you can start catalog synchronization, you should check that DFSMSrmm has all the authorization to read all ICF user catalogs including the tape volume catalog (TCDB). Normally, the TCDB is named SYS1.VOLCAT.VGENERAL.

When you enable catalog synchronization, DFSMSrmm synchronizes the DFSMSrmm control data set and available user catalogs. Enabling catalog synchronization is normally a one-time action. Once the catalog is synchronized, DFSMSrmm continually tracks and updates the catalog status. You enable catalog synchronization by specifying the DFSMSrmm EDGRMMxx OPTION CATSYSID operand described in the manual *DFSMSrmm Implementation and Customization Guide*, SC26-7405, and the chapter titled “Defining System Options:OPTION.”

In our example shown in Figure 8-3, we have three systems called SYS1, SYS2 and SYS3. System SYS1 and SYS2 have all ICF catalogs shared, and SYS3 is a single image without sharing any ICF catalog.

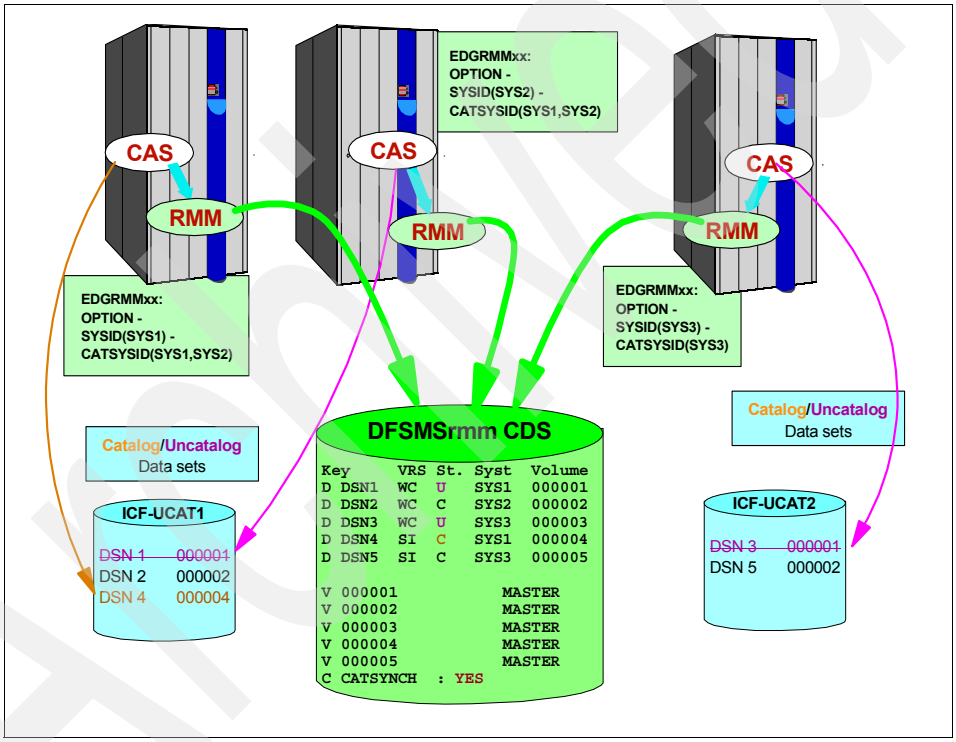


Figure 8-3 Catalog status tracking

Before you can run catalog synchronization

Before you start catalog synchronization in an environment with unshared catalogs, check for all current and previously used system IDs and include them in CATSYSID. Also, check for data sets with no create system ID because if there is no create system ID, you can lose data. There may be no create system ID as a result of the data set records being created during a conversion to RMM using

an older version of the conversion tools. Example 8-7 shows sample JCL that can be used to find out all previous and currently used create system IDs.

Recommendation: The create system ID field specifies the SMF system ID of the system where the data set was created.
When you run DFSMSrmm with catalogs that are not all fully shared, DFSMSrmm uses the create system ID of the first file on a volume to determine the system where the volume should return to scratch.

Example 8-7 Sample JCL to find out all used system IDs

```
//CRSYSID JOB (DE06304),SCHLUMBERGER,MSGLEVEL=1,MSGCLASS=H,
//             REGION=4M,NOTIFY=&SYSUID
//SORT     EXEC PGM=SORT
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMM.HSKP.BACKUP(0)
//SORTOUT  DD   DISP=(,CATLG,DELETE),
//         DCB=(LRECL=12,BLKSIZE=0,RECFM=VB),
//         DSN=RMM.HSKP.ALL.CATSYSID,
//         SPACE=(TRK,(1,1))
//SYSIN    DD   *
   SORT FIELDS=(73,8,CH,A),DYNALLOC
   OMIT COND=(5,1,CH,NE,C'D')
   OUTREC FIELDS=(1,4,73,8)
   SUM FIELDS=NONE
/*
```

Note: The DFSMSrmm CDS backup copy must be created by using EDGHSKP,PARM='BACKUP(AMS)'.

Figure 8-4 shows the result of your previous used SORT. You can see that you have an additional create system ID called TEST, and you have data set records without the create system information.

```
........
SYS1
SYS2
SYS3
TEST
```

Figure 8-4 List of all used system IDs

Note: Sometimes you get a blank or hex blank lines in the SORTOUT data set. That means you have data sets without create system ID information.

If you have data sets without the create system ID information, and all user catalogs are shared, you can use the JCL shown in Example 8-8 to find out all data sets without create system ID information. TSO RMM CHANGEDATASET subcommands are created to correct the errors.

Example 8-8 Sample JCL to find out all data sets without create system IDs info

```
//CRSYSID JOB (DE06304),SCHLUMBERGER,MSGLEVEL=1,MSGCLASS=H,
//             REGION=4M,NOTIFY=&SYSUID
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD   DUMMY
//SYSIN    DD   *
  DEL RMM.HSKP.WITHOUT.CATSYSID NONVSAM PURGE
  SET MAXCC=0
/*
//* ************************************************** *
//STEP02   EXEC PGM=EDGHSKP,PARM=(',RPTEXT,DATEFORM(J)')
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DSN=RMM.HSKP.MESSAGE.DSET,DISP=SHR
//XREPTEXT DD   DSN=RMM.HSKP.EXTRACTX,DISP=SHR
//* ************************************************** *
//SORT1    EXEC PGM=SORT
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMM.HSKP.EXTRACTX
//SORTOUT  DD   DISP=(,PASS,DELETE),
//         DSN=&&SORT1,
//         DCB=*.SORTIN,
//         SPACE=(CYL,(50,50))
//SYSIN    DD   *
     SORT FIELDS=(809,44,CH,A,915,4,CH,A),DYNALLOC
     INCLUDE COND=(809,44,CH,NE,C' ',AND,5,1,CH,EQ,C'X')
     OPTION VLSHRT
/*
//* ************************************************** *
//SORT2    EXEC PGM=SORT
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=(OLD,DELETE),DSN=&&SORT1
//SORTOUT  DD   DISP=(,CATLG,DELETE),
//         DCB=(LRECL=105,BLKSIZE=0,RECFM=VB),
//         DSN=RMM.HSKP.WITHOUT.CATSYSID,
//         SPACE=(TRK,(1,1))
//SYSIN    DD   *
     SORT FIELDS=(809,44,CH,A,915,4,CH,A),DYNALLOC
     INCLUDE COND=((809,44,CH,NE,C' '),AND,
                   869,8,CH,EQ,C'        ',OR,
                   869,8,CH,EQ,X'0000000000000000')
     OUTREC FIELDS=(1,4,C'          RMM CD ',809,44,
                    C' VOL(',9,6,C') SYSID(SYS1) SEQ(',
                    915,4,C') FORCE')
```

```
    OPTION VLSHRT
/*
```

Note: You only need to verify that the first data set on each volume has a create system ID, because that is the only data set that RMM EXPROC uses. This could result in a much shorter list of data sets to check and commands to be run.

Example 8-9 shows an example if you have catalogs that are not fully shared and how to find out all data sets without create system ID information residing in different volume pools. TSO RMM CHANGEDATASET subcommands are created with different create system ids to correct the errors. In the example below all volumes with a prefix of 5 and 9 are used on SY1 and all other volumes are used on SY3.

Example 8-9 Sample JCL to find out all data sets without create system ID info

```
//CRSYSID JOB (DEO6304),SCHLUMBERGER,MSGLEVEL=1,MSGCLASS=H,
//            REGION=4M,NOTIFY=&SYSUID
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD   DUMMY
//SYSIN    DD   *
  DEL RMM.HSKP.WITHOUT.CATSYSID.SY1 NONVSAM PURGE
  DEL RMM.HSKP.WITHOUT.CATSYSID.SY3 NONVSAM PURGE
  SET MAXCC=0
/*
//* ************************************************** *
//STEP02   EXEC PGM=EDGHSKP,PARM=(',RPTEXT,DATEFORM(J)')
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DSN=RMM.HSKP.MESSAGE.DSET,DISP=SHR
//XREPTEXT DD   DSN=RMM.HSKP.EXTRACTX,DISP=SHR
//* ************************************************** *
//SORT1    EXEC PGM=SORT
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=SHR,DSN=RMM.HSKP.EXTRACTX
//SORTOUT  DD   DISP=(,PASS,DELETE),
//         DSN=&&SORT1,
//         DCB=*.SORTIN,
//         SPACE=(CYL,(50,50))
//SYSIN    DD   *
    SORT FIELDS=(809,44,CH,A,915,4,CH,A),DYNALLOC
    INCLUDE COND=(809,44,CH,NE,C' ')
    OPTION VLSHRT
/*
//* ************************************************** *
//SORT2    EXEC PGM=SORT
//SYSOUT   DD   SYSOUT=*
```

```
//SORTIN   DD   DISP=(OLD,PASS),DSN=&&SORT1
//SORTOUT  DD   DISP=(,CATLG,DELETE),
//         DCB=*.SORTIN,
//         DSN=RMM.HSKP.WITHOUT.ATSYSID.SY1,
//         SPACE=(TRK,(10,1))
//*        ************************************************
//*        * include all volumes starting with 5 and 9     *
//*        ************************************************
//SYSIN    DD   *
     SORT FIELDS=(809,44,CH,A,915,4,CH,A),DYNALLOC
     INCLUDE COND=((9,1,CH,EQ,C'5',OR,9,1,CH,EQ,C'9'),AND,
                  (869,8,CH,EQ,C'        ',OR,
                   869,8,CH,EQ,X'0000000000000000'))
     OPTION VLSHRT
/*
//* ************************************************** *
//SORT3    EXEC PGM=SORT
//SYSOUT   DD   SYSOUT=*
//SORTIN   DD   DISP=(OLD,DELETE),DSN=&&SORT1
//SORTOUT  DD   DISP=(,CATLG,DELETE),
//         DCB=*.SORTIN,
//         DSN=RMM.HSKP.WITHOUT.CATSYSID.SY3,
//         SPACE=(TRK,(1,1))
//*        ************************************************
//*        * exclude all volumes starting with 5 and 9     *
//*        ************************************************
//SYSIN    DD   *
     SORT FIELDS=(809,44,CH,A,915,4,CH,A),DYNALLOC
     INCLUDE COND=((9,1,CH,NE,C'5',AND,9,1,CH,NE,C'9'),AND,
                  (869,8,CH,EQ,C'        ',OR,
                   869,8,CH,EQ,X'0000000000000000'))
     OPTION VLSHRT
/*
```

Figure 8-5 shows the created TSO RMM CHANGEDATASET subcommands to correct all data set records not having the creating-system information available.

```
RMM CD DE49817.SYS1.DUMP.D000630 VOL(SD2180)   -
       SYSID(SYS1) SEQ(   1) FORCE
```

Figure 8-5 TSO RMM subcommands to update data set records

Figure 8-10 shows sample JCL to execute all TSO RMM CHANGEDATASET subcommands created in the previous JOB.

Archived

Example 8-10 Sample JCL to execute all TSO RMM CHANGEDATASET command

```
//CRSYSID JOB (DEO6304),SCHLUMBERGER,MSGLEVEL=1,MSGCLASS=H,
//             REGION=4M,NOTIFY=&SYSUID
//IKJEFT01 EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
```

Note: Add the PROF NOPREFIX command in front so you do not have to specify the data set names in single quotation marks.

Important: If you have data set records without creating system information, you can lose data.

Example 8-11 shows you the different EDGRMMnn PARMLIB members for the systems we are describing.

Example 8-11 EDGRMMnn PARMLIB members

```
System SYS1:

   OPTION OPMODE(WARNING)       -
          .....
          SYSID(SYS2)           -
          CATSYSID(SYS1,SYS2)   -
          ....

System SYS2:

   OPTION OPMODE(WARNING)       -
          .....
          SYSID(SYS1)           -
          CATSYSID(SYS1,SYS2)   -
          ....

System SYS3:

OPTION OPMODE(WARNING)       -
       .....
       SYSID(SYS3)           -
       CATSYSID(SYS3)        -
       ....
```

In our example (shown in Figure 8-6), we have three systems called SYS1, SYS2 and SYS3. The first job can run on any of the systems, and then run the catalog

synchronization on system SYS1 or SYS2 system, and when finished, run the catalog syncronization on SYS3. If all jobs are run without any errors, you can enable the catalog syncronization. The job to enable catalog synchronization can run on any of the systems.

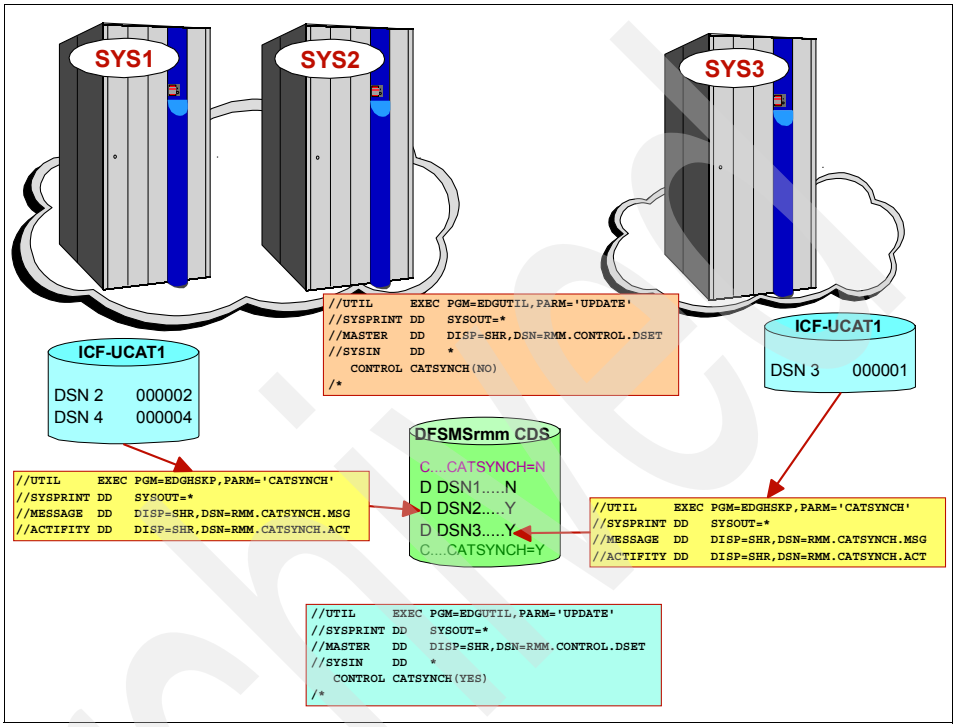


Figure 8-6 Enabling catalog synchronization

Disable catalog synchronization

Use CATSYNCH(NO) to clear the last synchronization date and time stored in the DFSMSrmm control record. After a conversion, the catalog synchronization is normally not switched on. Use the JCL shown in Example 8-12 to disable catalog synchronization.

Example 8-12 JCL to disable catalog synchronization

```
//CATSYNCO JOB (DEO6304),SCHLUMBERGER,MSGLEVEL=1,MSGCLASS=H,
//            REGION=4M,NOTIFY=&SYSUID
//UTIL     EXEC PGM=EDGUTIL,PARM='UPDATE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.CONTROL.DSET
//SYSIN    DD   *
```

```
   CONTROL CATSYNCH(NO)
/*
```

You can check that the catalog synchronization is disabled by using the TSO RMM subcommand LISTCONTROL as shown in Example 8-13.

Example 8-13 RMM LISTCONTROL subcommand

```
TSO RMM LISTCONTROL
```

Example 8-14 is the result of the TSO RMM LISTCONTROL subcommand. The last catalog synchronization date and time fields should be blank since the function is disabled.

Example 8-14 Output of the TSO RMM LISTCONTROL subcommand

```
Control record:
Type = MASTER    Create date = 1997/114    Create time = 14:28:32
Journal Utilization =  0%  (75% threshold)      STATUS: = DISABLED
CDS:    Utilization =  75%
Exit status:                              Options:
  EDGUX100 = ENABLED                        Stacked Volumes         = NONE
  EDGUX200 = NONE                           Extended Bin            = DISABLED
Last backup:                              Last expiration processing:
  Date = 2003/252    Time = 16:58:03        Date = 2003/252    Time = 16:55:29
Last journal backup:                      Last store update:
  Date = 2003/252    Time = 16:58:03        Date = 2003/184    Time = 17:47:19
Last report extract:                      Last VRS processing:
  Date = 2003/196    Time = 15:02:00        Date = 2003/252    Time = 16:55:29
Last scratch procedure:                   Last Catalog synchronize:
  Date = 2000/006    Time = 05:10:23        Date =             Time =
Rack numbers       = 2842                 Empty racks        = 1800
LOCAL store bins   = 244                  Empty LOCAL bins   = 244
DISTANT store bins = 200                  Empty DISTANT bins = 120
REMOTE store bins  = 999                  Empty REMOTE bins  = 999
Control functions in progress:
Backup         = N  Restore        = N
Verify         = N  Expiration     = N
Report Extract = N  Disaster Store = N
VRS            = N  Synchronize    = N

READY
```

Allocating a larger message data set

You must allocate a larger DFSMSrmm message data set, because the catalog synchronization process will write a record to the DFSMSrmm message file for

Archived

each data set that is updated in the DFSMSrmm control data set. You can use the JCL shown in Example 8-15 to allocate a larger message data set.

Example 8-15 JCL to allocate a larger DFSMSrmm message data set

```
//CATSYNC1 JOB (DEO6304),SCHLUMBERGER,MSGLEVEL=1,MSGCLASS=H,
//            REGION=4M,NOTIFY=&SYSUID
//IEFBR14  EXEC PGM=IEFBR14
//MESSAGE  DD   DISP=(,CATLG,DELETE),
//         DCB=(LRECL=133,BLKSIZE=6118,RECFM=FBA),
//         DSN=RMM.HSKP.MESSAGE.CATSYNC,
//         SPACE=(TRK,(100,100))
```

Journal considerations

The catalog synchronization process will also generate many records in the journal data set. Prepare for this by allocating a large journal data set, or process the catalog synchronization run with the journal data set disabled. Refer to 8.3.2, “Disabling the journal data set” on page 347 for more information about disabling the DFSMSrmm journal data set.

JCL for catalog synchronization

To request catalog synchronization, specify the CATSYNCH parameter. Before you start this function, check the size of your message data set. The DFSMSrmm subsystem writes a message to this data set for each record that is updated in this run.You must submit a job with JCL similar to that in Example 8-16 on each catalog plex; in our example, one on SYS1 or SYS2 and one on SYS3. You do not have to stop tape processing because this function can run in parallel to your normal jobs using tape drives. DFSMSrmm always retrieves catalog information to determine the catalog status of a data set.

Example 8-16 JCL for catalog synchronization

```
System SYS1 or SYS2 and System SYS3:
   //CATSYNC1 JOB (DEO6304),SCHLUMBERGER,MSGLEVEL=1,MSGCLASS=H,
   //            REGION=4M,NOTIFY=&SYSUID
   //HSKP     EXEC PGM=EDGHSKP,
   //         PARM='CATSYNCH'
   //MESSAGE  DD   DSN=MESSAGE.FILE.NAME.CATSYNC,DISP=SHR
   //ACTIVITY DD   DSN=ACTIVITY.FILE.NAME,DISP=SHR
```

Look to the message data set, and you can see all data sets that are updated in the DFSMSrmm control data set by this run. Figure 8-17 shows you an example.

Note: Rerun the catalog synchronization if you are manually adding volumes in MASTER or USER status with data set information, or you are changing your catalog environment (MERGECAT, IMPORT, EXPORT).

Example 8-17 Message data set after catalog synchronization

```
EDG6001I INVENTORY MANAGEMENT STARTING ON 04/11/2002 AT 10:58:34 -
         PARAMETERS IN USE ARE
         DATEFORM(E),CATSYNCH
EDG2309I THE PARMLIB OPTIONS CURRENTLY IN USE ARE
         VRSEL(NEW)
         VRSJOBNAME(2)
         VRSMIN(10,FAIL)
         VRSCHANGE(INFO)
         SMSTAPE(PURGE(ASIS) UPDATE(EXITS,SCRATCH,COMMAND))
         CATRETPD(24)
         UNCATALOG(Y)
         TPRACF(A)
         NOTIFY(Y)
         SYSID(EGZB)
         CATSYSID(*)
EDG2234I DFSMSrmm CDS CATALOG STATUS UNKNOWN FOR SCHLUM.RMMDEMO.FILE4.VOL23
         VOLUM.RMMDEMO.FILE4.VOL23 VOLUME SCO010 FILE 3 SYNCHRONIZED TO
         CATALOG STATUS
EDG2234I CONT:-  YES
....
EDG2234I DFSMSrmm CDS CATALOG STATUS YES FOR WWZ038.GSD1997A.DUMP VOLUME
         SCO005 FILE 1 SYNCHRONIZED TO CATALOG STATUS NO
EDG2234I DFSMSrmm CDS CATALOG STATUS YES FOR WWZ038.GSD1997A.DUMP VOLUME
         SCO011 FILE 1 SYNCHRONIZED TO CATALOG STATUS NO
....
EDG2234I DFSMSrmm CDS CATALOG STATUS NO FOR WWZ038.SAVE.ZOS120.DU01A426
         VOLUME Q17062 FILE 1 SYNCHRONIZED TO CATALOG STATUS YES
EDG2234I DFSMSrmm CDS CATALOG STATUS NO FOR WWZ038.SAVE.ZOS120.DU01A483
         VOLUME Q17030 FILE 1 SYNCHRONIZED TO CATALOG STATUS YES
EDG2307I INVENTORY MANAGEMENT TASK CATSYNCH COMPLETED SUCCESSFULLY
EDG6901I UTILITY EDGHSKP COMPLETED WITH RETURN CODE 0
```

Enable catalog synchronization

You must use CATSYNCH(YES) to indicate to DFSMSrmm that catalogs are synchronized when you run DFSMSrmm with unshared catalogs. Run EDGUTIL with PARM=UPDATE parameter specified and CONTROL CATSYNCH(YES) to set the last catalog synchronization date and time in the DFSMSrmm control data set control record. Example 8-18 shows you sample JCL to enable catalog synchronization. You must run this job only once.

Archived

Example 8-18 JCL to enable catalog synchronization

```
//UTIL     EXEC PGM=EDGUTIL,PARM='UPDATE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//SYSIN    DD   *
   CONTROL CATSYNCH(YES)
/*
```

Check that the catalog synchronization is enabled successfully by using the TSO RMM subcommand LISTCONTROL as shown in Example 8-19.

Example 8-19 RMM LISTCONTROL subcommand

```
TSO RMM LISTCONTROL
```

Example 8-20 is the output of the TSO RMM LISTCONTROL subcommand after enabling catalog synchronization. Note the last catalog synchronization date and time fields now contain data.

Example 8-20 RMM LISTCONTROL output

```
Control record:
Type = MASTER    Create date = 1997/114    Create time = 14:28:32
Journal Utilization =  0%  (75% threshold)      STATUS: = DISABLED
CDS:    Utilization =  75%
Exit status:                                Options:
  EDGUX100 = ENABLED                          Stacked Volumes         = NONE
  EDGUX200 = NONE                             Extended Bin            = DISABLED
Last backup:                                Last expiration processing:
  Date = 2003/252    Time = 16:58:03          Date = 2003/252    Time = 16:55:29
Last journal backup:                        Last store update:
  Date = 2003/252    Time = 16:58:03          Date = 2003/184    Time = 17:47:19
Last report extract:                        Last VRS processing:
  Date = 2003/196    Time = 15:02:00          Date = 2003/252    Time = 16:55:29
Last scratch procedure:                     Last Catalog synchronize:
  Date = 2000/006    Time = 05:10:23          Date =  2002/193   Time = 03:17:05
Rack numbers       = 2842                   Empty racks        = 1800
LOCAL store bins   = 244                    Empty LOCAL bins   = 244
DISTANT store bins = 200                    Empty DISTANT bins = 120
REMOTE store bins  = 999                    Empty REMOTE bins  = 999
Control functions in progress:
Backup         = N  Restore        = N
Verify         = N  Expiration     = N
Report Extract = N  Disaster Store = N
VRS            = N  Synchronize    = N
READY
```

Archive

Enable journal data set (optional)

When the catalog synchronization is successfully finished, the journal data set should be activated again. Refer to 8.3.3, “Enabling the journal data set” on page 349 for more information about activating the DFSMSrmm journal data set.

Note: If the journal was previously used, you must delete it and allocate a new one, because the CDS and JOURNAL control information no longer match.

8.1.11 Check interfaces

Before DFSMSrmm can run in parallel to TLMS, there is some additional information that you must consider about interfaces.

SMS interface

If you have an IBM 3494 or an IBM 3495 Automated Tape Library Dataserver, you must ensure that it works correctly in the DFSMSrmm environment. You must install the DFSMSrmm-supplied OAM exits for the DFSMSrmm version and provide the correct input to the extract program for those volumes that belong to the tape library.

You must define an DFSMSrmm location with the same name as the tape library name, using a location type of ATL. The name of the ATL is defined in your SMS environment.

BTLS interface

If your installation uses BTLS to drive IBM 3494 and IBM 3495 Automated Tape Library Dataserver operations, you will need to notify BTLS when a volume must be returned to scratch status. In your current environment, this is done through a batch job that gets information from your TLMS database, and then updates the BTLS catalog. In DFSMSrmm, you can use a simple REXX CLIST that synchronizes the DFSMSrmm CDS with the BTLS catalog.

Short-on-scratch processing

DFSMSrmm offers an integrated SMS interface that updates the TCDB automatically. In addition, you can free volumes that are in PENDING RELEASE status when a short-on-scratch condition is detected inside the automated tape libraries without manual intervention. Use the SCRATCHPROC(EDGXPROC) PARMLIB option, where EDGXPROC is the name of the procedure DFSMSrmm starts to replenish scratch volumes in the automated tape libraries.

TSO and ISPF interface

DFSMSrmm has a simple TSO interface that uses a set of powerful commands to drive all tape activities.

There is also an ISPF interface that offers menus for end users, librarians, administrators, and support personnel. You can choose a full DFSMSrmm dialog or a local defined dialog.

You can tailor the correct ISPF dialog for each user category and assign each the needed access to DFSMSrmm. See the *DFSMSrmm Implementation and Customization Guide*, SC26-7405 for more information.

Other interfaces

DFSMSrmm can use any DFSMSrmm command in batch mode, using the IKJEFT01 or IKJEFT1A batch TSO interface. It can also process any DFSMSrmm command through the DFSMSrmm API. To use the DFSMSrmm API, you need to code in the High Level Assembler language. A preferred method is to use the DFSMSrmm commands from a REXX environment where you can retrieve information back as variables.

If you have any of those interfaces running in your current tape environment, you must change them to reflect the equivalent function in DFSMSrmm.

STK interfaces

Following are the STK recommendations if you have a STK VSM installed.

Protect MVC volumes using RACF

These are the steps:

1. Define a profile in the TAPEVOL class for Multi-Volume Cartridge (MVC) as shown Example 8-21.

Example 8-21 Add a TAPEVOL profile

```
RDEFINE TAPEVOL CVC024 UACC(NONE)
```

2. Allow user ID VSM8HSC (STC user ID) update access to MVC CVC024 as shown in Example 8-22.

Example 8-22 Permit HSC to MVCs

```
PERMIT CVC024 CLASS(TAPEVOL) ACCESS(UPDATE) ID(VSM8HSC)
```

3. Allow STC the use of LISTVOL and CHANGEVOL commands.

Archived

This is done by permitting user ID VSM8HSC (STC userID) control access as shown in Example 8-23.

Example 8-23 Permit HSC to RMM resources

```
PERMIT STGADMIN.EDG.MASTER ACCESS(CONTROL) ID(VSM8HSC)
```

See *Security Server RACF Security Administrator's Guide*, SA22-7683 for more information about the resources defined in the RACF FACILITY class.

4. Protect VTCS commands.

   Define a profile in the OPERCMDS class for all VTCS commands. You can use a RACF RDEFINE OPERCMDS as shown in Example 8-24.

Example 8-24 ADD an OPERCMDS resource

```
RDEFINE OPERCMDS subsystemname.VT UACC(NONE)
```

5. Allow user SAM15 update access to all VTCS commands you have previously defined as shown in Example 8-25.

Example 8-25 PERMIT VSM to OPERCMDS

```
PERMIT subsystemname.VT CLASS(OPERCMDS) ID(SAM15) ACC(UPDATE)
```

Caution: VTCS has an automatic interface to notify DFSMSrmm when a VTV becomes scratch but DFSMSrmm does not notify VTCS when unscratching a VTV by changing the volume status to non-scratch. If this is done, you must also run the HSC SLUADMIN utility to unscratch the VTV for VTCS. Otherwise, you may encounter a mount failure when VTCS attempts to service a scratch mount request.

8.2 Parallel processing

The timing of the DFSMSrmm update to its CDS during OPEN processing is such that any volumes rejected by the vendor system should not result in updates to the DFSMSrmm CDS.

Also, the timing of the DFSMSrmm reject of a tape is such that recording by TLMS may not occur if DFSMSrmm rejects a volume. Therefore, parallel running problems are unlikely to be the result of volume rejection and control by one system or the other. More likely, the differences in inventory management functions such as expiration and retention control cause the problems. DFSMSrmm's uncataloging of tape data sets can cause the other vendor system to return a tape to scratch, and vice versa.

Timing of inventory management and batch runs may vary, causing volumes to return to scratch at different times, or to move between storage locations at different times. This can cause problems if both systems are in protect mode at the same time. If DFSMSrmm is in record-only or warning mode, the differences should not be a significant problem because DFSMSrmm continues to allow tapes to be used, and keeps track of the latest tape information.

Note: When running in parallel, you should prevent DFSMSrmm from uncataloging tape data sets, but continue to run both sets of inventory management. This can be accomplished by setting the PARMLIB option to **UNCATALOG(N)** in member EDGRMMxx.

Another method to consider for conversion is to run both DFSMSrmm and the other vendor product in protect mode in parallel, but use different ranges of volumes managed by each system. You can use the exits available in each product (EDGUX100 for DFSMSrmm) to control which volumes they manage and which volumes they are to ignore or treat as foreign. In time, as volumes return to scratch, or as new applications are added, you can move more and more volumes to DFSMSrmm control.

8.3 Starting tape and inventory management

When you have checked the EDGRMMxx PARMLIB member, you are ready to start DFSMSrmm. To start the DFSMSrmm subsystem, issue the command shown in Example 8-26.

Example 8-26 Starting the DFSMSrmm subsystem

```
S DFRMM,M=xx
```

If you have not enabled the DFSMSrmm subsystem at IPL time, you receive the message shown Example 8-27.

Example 8-27 Message EDG0103D

```
EDG0103D DFSMSrmm SUBSYSTEM INTERFACE IS INACTIVE -
         ENTER "IGNORE","CANCEL" or "RETRY"
```

Reply RETRY to this message. In response, DFSMSrmm initializes its subsystem interface. If you reply IGNORE, the DFSMSrmm tape recording function is not activated.

Before you can start your tape activities, run inventory management as follows:

1. Run vital record processing first before expiration and storage location management processing to identify which volumes to retain and where volumes should be moved, based on vital record specifications.
2. Run expiration processing to identify those volumes not required for vital records that are ready to expire. During expiration processing, release actions for volumes are noted.
3. Run storage location management to assign shelf locations in storage locations to volumes that are being stored as vital records.
4. You must run storage location management processing after vital record processing has been successfully run, but not necessarily in the same run of EDGHSKP. Storage location management assigns the exact shelf location to be used for the volume while it is in the storage location.

8.3.1 First run of EDGHSKP with PARM(VRSEL)

The first run of EDGHSKP vital record selection processing will run longer than all subsequent runs of EDGHSKP. The reason for this discrepancy in runs is that DFSMSrmm must update all volume and data set records to set the correct vital record status and retention information.

Note: The first run will also generate many records in the journal data set. Prepare for this by allocating a large journal data set, or process the first EDGHSKP PARM(VRSEL) run with the journal data set disabled.

In the EDGRMMxx PARMLIB member you can define the JOURNALFULL parameter. If it is not specified, the value defaults to 75, which means that when the journal data set is 75% filled, the warning message shown in Example 8-28 is written to the system log.

Example 8-28 Message EDG2107E

```
EDG2107E JOURNAL THRESHOLD REACHED -
         JOURNAL IS percentage_value% FULL.
          tracks TRACKS (kilobytesK) AVAILABLE
```

If you set the value JOURNALFULL(0), DFSMSrmm will not issue any warning messages. See the EDGRMMxx PARMLIB member description in 8.1.3, “EDGRMMxx - DFSMSrmm options” on page 311 for an explanation of the JOURNALFULL and the relation to the BACKUPPROC parameter.

When a journal full situation occurs, the messages shown in are written to the system console.

Example 8-29 Messages EDG2104E and EDG2103D

```
EDG2104E JOURNAL FILE IS FULL - SCHEDULE CONTROL DATA SET
         BACKUP TO CLEAR IT.

EDG2103D PERMANENT JOURNAL ERROR - REPLY 'R' TO RETRY,
         'I' TO IGNORE, 'D' TO DISABLE, OR 'L' TO LOCK.
```

When these messages appear, reply to EDG2103D with **'D'**, to disable the journal data set. The rest of the EDGHSKP run will then continue without journaling. If the inventory management process is successfully finished you can enabling your journal by using the EDGHSKP PARM(BACKUP) function.

8.3.2 Disabling the journal data set

Another possibility is to do the first run of EDGHSKP with the journal disabled from the beginning. The possibility of disabling the journaling depends on how the journal data set is defined in the DFSMSrmm procedure and in the PARMLIB member EDGRMMxx.

If the journal data set is not defined in the DFSMSrmm procedure, and is defined in the PARMLIB member EDGRMMxx, it is possible to do the following steps to perform a EDGHSKP PARM(VRSEL) run with the journaling disabled:

1. Create a new member EDGRMMyy by copying the existing member EDGRMMxx in PARMLIB.
2. In the new member EDGRMMyy remove the JRNLNAME statement in the OPTION parameter.
3. Restart DFSMSrmm with the new EDGRMMyy member, by issuing the operator MODIFY command shown in Example 8-30.

Example 8-30 Restarting the DFSMSrmm subsystem

```
F DFRMM,M=yy
```

4. When DFSMSrmm is restarted without the journal data set you receive the messages shown in Example 8-31 in your system log.

Example 8-31 Messages EDG0204I, EDG0122I and EDG0105I

```
EDG0204I DFSMSrmm BEING INITIALIZED FROM MEMBER EDGRMMyy IN SYS1.PARMLIB
EDG0122I NO JOURNAL FILE ALLOCATED - JOURNALING DISABLED
EDG0105I DFSMSrmm SUBSYSTEM INITIALIZATION COMPLETE
```

Archived

You can also check the status of the journal data set by issuing the command shown in Example 8-32.

Example 8-32 LISTCONTROL to list options

```
RMM LISTCONTROL OPTION
```

The output of the LISTCONTROL command is shown in Example 8-33.

Example 8-33 LISTCONTROL OPTION output

```
Control record:
Type = MASTER    Create date = 1997/114    Create time = 14:28:32
Journal Utilization =  0%  (75% threshold)      STATUS: = DISABLED
CDS:    Utilization =  75%
Exit status:                                Options:
  EDGUX100 = ENABLED                          Stacked Volumes         = NONE
  EDGUX200 = NONE                             Extended Bin            = DISABLED

System options:
PARMLIB Suffix  = 00
Operating mode  = W    Retention period: Default = 5       Maximum = NOLIMIT
                                         Catalog = 12    hours
Control data set name      = RMM.PROD.CDS
Journal file data set name =
Journal threshold          = 75%
Catalog SYSID   = *
Scratch procedure name     = EDGXPROC
Backup procedure name      = EDGBKUP
IPL date check  = N        Date format    = J       RACF support   = A
SMF audit       = 248      SMF security   = 249     CDS id         = PROD
MAXHOLD value   = 100      Lines per page = 54      System ID      = EGZB
BLP             = RMM      TVEXT purge    = RELEASE Notify         = Y
Uncatalog       = Y        VRS job name   = 2       Message case   = M
MASTER overwrite= LAST     Accounting     = J       VRS selection  = NEW
VRS change      = INFO     VRSMIN action  = FAIL    VRSMIN count   = 10
Disp DD name    = LOANDD   Disp msg ID    = EDG4054I
Retain by       = VOLUME   Move by        = VOLUME  CMDAUTH Owner  = NO
PREACS          = NO       SMSACS         = NO      CMDAUTH Dsn    = YES
Reuse bin       = CONFIRMMOVE
PDA: OFF
 Block count    = 255        Block size     = 27       Log            = OFF
SMSTAPE:
 Update scratch = YES      Update command = YES     Update exits   = YES
 Purge          = ASIS

READY
```

The `Journal file data set name` field should be blank when the journal data set is disabled.

Archived

5. DFSMSrmm is now running with the journal data set disabled and the EDGHSKP PARM(VRSEL) job can start. Running EDGHSKP without journaling also reduces the execution time.

8.3.3 Enabling the journal data set

When the EDGHSKP run is complete, the journal data set should be activated again. To activate the journal data set do the following:

1. Restart DFSMSrmm with the EDGRMMxx PARMLIB member, with the command shown in Example 8-34.

Example 8-34 Restarting the DFSMSrmm subsystem

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

2. Backup the DFSMSrmm CDS and the journal data set, with the EDGHSKP PARM(BACKUP) utility.

DFSMSrmm journal data set is now active, nullified, and synchronized with the DFSMSrmm CDS after the backup.

8.4 Validating DFSMSrmm operation

In this section we show how to validate the DFSMSrmm operation. During the validation phase of the conversion process, you must create reports to compare and verify the correct conversion of the database and policies. We will explain how to obtain reports that are useful for product comparisons.

8.4.1 Data set name list of non-scratch volumes

In both tape management systems, you should produce a list of the data sets for every non-scratch volume. This list should include:

- Data set name
- VOLSER
- Retention date or equivalent in the vendor product
- Assigned or creation date or equivalent
- Job name
- Volume sequence number
- Data set sequence number

Archived

You can get this information with the DFSMSrmm extract file as input, and then use DFSMSrmm delivered REXX execs and JCL to create the report you need. Creating reports is described in Chapter 12, “Reporting on your DFSMSrmm environment” on page 485.

For TLMS, use the TMSRPT2 utility to produce an equivalent report.

When you have a report from DFSMSrmm and a report in identical format from your existing product, you can compare the two report files using a compare utility such as ISPF SuperC.

8.4.2 Scratch volume lists

A list of scratch volumes should be produced for both products, ordered by VOLSER, and then compared using the SuperC utility.

To obtain this list from DFSMSrmm, you can use the command in Example 8-35.

Example 8-35 SEARCHVOLUME to list all volumes in status SCRATCH

```
RMM SV * STATUS(SCRATCH) OWNER(*) LIMIT(*)
```

Using the IKJEFT01 or IKJEFT1A TSO batch interface program you can store the results of this command in a data set referenced by the SYSTSPRT DD statement. Then you can compare the result with the list obtained with a similar utility or command of TLMS.

8.4.3 Inventory by location

You can produce an inventory of the volumes present in every storage location defined and then compare it with an equivalent inventory of the vendor product volumes. Use the DFSMSrmm command shown in Example 8-36.

Example 8-36 SEARCHVOLUME to list a volumes in a specific location

```
RMM SV * LOCATION(loc_name) STATUS(MASTER) OWNER(*) LIMIT(*)
```

In this command, **`loc_name`** is the storage location name you want to use for the comparison.

Alternatively, you can create DFSMSrmm REPORT06 described in *DFSMSrmm Reporting* to list the volumes in every storage location sorted by VOLSER.

Archived

8.4.4 Multifile multi-volume reports

A multifile multi-volume report is created using DFSMSrmm REPORT11 described in *DFSMSrmm Reporting*. The multifile multi-volume report includes the following information:

- VOLSER
- Volume sequence
- Data set sequence
- Data set name
- Expiration date
- First volume
- Previous volume
- Next volume
- Creating user ID
- Creating job name
- Creating date
- Creating time

The report is sorted by the first file on the first volume, volume sequence, and file sequence.

8.4.5 Comparing results

You can have differences between TLMS and DFSMSrmm in:

- Storage location bins
- Volume expiration date
- Logical and physical file sequence number
- multi-volume chain
- Job name creation
- Scratch lists

You must compare the results of DFSMSrmm inventory management processing with the equivalent processing in your existing product. You can compare the results from the reports produced using a product like ISPF/PDF SuperC.

When you have compared the results of processing of DFSMSrmm and your existing system and have verified that there are no differences, or that the differences are minimal, understood, and acceptable, then you are ready to cut over to production.

Archived

Storage location bins

Storage location management processing assigns a shelf location, known as the bin number, to volumes moving to and among storage locations and sets a destination location. Storage location management processing must be run to manage volume movement, whether or not the moves involve storage locations.

The algorithm used by DFSMSrmm for bin assignment can result in more bin numbers being required than with your existing system. DFSMSrmm processes volumes in ascending VOLSER sequence, and for each volume can confirm a move, freeing a bin number, and initiate a move, assigning a bin number. A volume later in sequence can use the freed bin number. Free bin numbers are assigned in ascending bin number sequence.

DFSMSrmm does not assign a bin number to more than one volume at a time. Bin numbers are freed only when the move from the location has been confirmed and the volume updated in the DFSMSrmm CDS.

Volume expiration date

Any non-scratch volume defined to DFSMSrmm has an expiration date indicating when the volume is to be considered for release. This date can be set for the volume by:

- An expiration date or retention period, given in the JCL by the user when writing a data set to the volume
- The DFSMSrmm GETVOLUME command, when a user manually requests a scratch volume, or when information about the volume was manually added or changed
- A default retention period for data sets set by your installation with the RETPD option in EDGRMMxx in SYS1.PARMLIB.

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

You can retain a volume beyond its expiration date through a VRS. You can also override an expiration date and release a volume manually before the expiration date is reached.

The volume expiration date is the highest expiration date of any one data set on this volume. The first data set on a volume has no priority to keep or move a volume.

Archived

Note: DFSMSrmm can handle mixed expiration dates without any problems. This means you can keep a volume by a VRS and a retention period. For more information see the *DFSMSrmm Guide and Reference*, SC26-7404, SC26-7404.

In addition to the volume expiration date, DFSMSrmm calculates and saves a retention date for each data set and volume. The retention date probably is a more accurate match to the expiration or keep date known to your existing system.

Logical file 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 1; the maximum allowable decimal value is 9999.

Physical file sequence number

Specifies the relative position of the data set on the volume. The minimum allowable decimal value is 1; the maximum allowable decimal value is 9999. When you add a data set that is not the first data set on a volume, the preceding data sets on the volume must already be defined to DFSMSrmm.

Your existing system records only the logical file sequence number.

Multi-volume chain

DFSMSrmm manages individual data sets using VRS definitions. Each data set on a volume or multi-volume set can have different retention criteria. DFSMSrmm does not attempt to manage the multi-volume set based on a controlling file, but allows each volume to be retained separately based on the data sets it contains.

As a result, DFSMSrmm can expire and return to scratch one or more volumes from a multi-volume set while still retaining the other volumes that have data sets managed by VRS, or that have later expiration dates.

You may have to add extra VRS definitions so that all data sets in a multi-volume set are managed using common retention criteria if the data sets on the released volumes should also be retained.

Archive

Job name creation

You can have differences in your job name creation because, in the DFSMSrmm volume record, the JOBNAME is the name of the job that created the first data set on the volume and, in the data set record, the JOBNAME specifies the name of the job that created the data set.

Comparing scratch lists

You can have differences in your scratch lists because of the PENDING RELEASE status used by DFSMSrmm for volumes that have reached their expiration date and are not retained by VRS.

The expiration of tapes can be set to immediate in VRS definitions, so some categories of tapes can be released immediately in the first EDGHSKP EXPROC run. However there can still be differences in scratch lists between both tape management systems.

You also can have differences in your scratch list because of volumes kept by an External Data Manager. The volume status of a volume released by DFSMShsm is set to pending release by the EDGDFHSM built-in interface call and changed to scratch status by the first expiration processing. There is no option for extending the retention date when a volume is released in this way.

8.4.6 Checking WTORs and messages

We recommend that you have the system operator and automated operation program check for and answer specific WTORs quickly. If the message is still open, all tape processing is stopped.

You can check the DFSMSrmm messages in your JOB LOG and SYSTEM LOG only if DFSMSrmm is running in `WARNING` mode. Check all warning messages and correct the error, if necessary.

For more information, see *z/OS MVS System Messages, Vol. 5 (EDG-GFS)*, SA22-7635.

8.5 Keeping databases synchronized

If you run DFSMSrmm in parallel to TLMS, try to avoid manual changes. If you must change expiration dates, add or delete volumes, or release volumes in your old environment, you must make the same changes in the DFSMSrmm CDS. Only if you update both tape management systems can you compare them successfully.

Archived

If you change DFSMSrmm policies, PARMLIBs, or exits, keep track of them so that they are automatically included if a reconversion is performed. Make all changes through batch and save the TSO DFSMSrmm subcommands.

Archived

Archived

9

Cutover to production

This chapter explains how to cut over to production.

Objectives of this chapter:

- List and explain all tasks required to run DFSMSrmm in production with minimum risk.
- Get DFSMSrmm into production and deactivate vendor product.
- Ensure that correct tasks are executed at the correct time.
- Ensure that cutover will be performed only when ready.

In this chapter we describe the steps you must follow to set up your DFSMSrmm for production cutover and remove the vendor product.

At the end of this chapter you will have DFSMSrmm installed in production and will have removed TLMS. You will be ready to exploit additional DFSMSrmm functions or continue business as usual.

© Copyright IBM Corp. 2004. All rights reserved.

9.1 EDGRMMxx PARMLIB options

In this section we describe the steps you must follow to set up your system for production cutover and remove TLMS.

Your DFSMSrmm system is probably running in *record-only* or *warning* mode, because during the parallel running and validation phase you had to check for differences and verify the correct conversion of the vendor database.

The next time you start or restart DFSMSrmm, you must change several PARMLIB options to use DFSMSrmm with the full set of functions active. Example 9-1 shows a sample EDGRMMxx PARMLIB member for running DFSMSrmm in *protect* mode. Parameters that affect protect mode running are highlighted.

Example 9-1 Sample EDGRMMnn PARMLIB member for PROTECT mode

```
OPTION    OPMODE(P)                 /* Protect Mode              */ -
          ACCOUNTING(J)             /* Accounting from JOB       */ -
          BACKUPPROC(EDGBKUP)       /* Name of BACKUP-proc       */ -
          BLP(RMM)                  /* DFSMSrmm controls BLP     */ -
          CATRETPD(0012)            /* catalog retention         */ -
          CATSYSID(*)               /* all catalogs shared       */ -
          CDSID(PROD)               /* control data set id       */ -
          COMMANDAUTH(OWNER)        /* type of authorization     */ -
          DATEFORM(J)               /* Date format               */ -
          DISPDDNAME(LOANDD)        /* DISP ctrl  DD card        */ -
          DISPMSGID(EDG4054I)       /* DISP message number       */ -
          DSNAME(RMM.PROD.CDS)      /* CDS data set name         */ -
          IPLDATE(N)                /* IPL date checking         */ -
          JRNLNAME(RMM.PROD.JRNL)   /* JRNL data set name        */ -
          JOURNALFULL(75)           /* Percentage JRNL full      */ -
          LINECOUNT(054)            /* Lines per page            */ -
          MASTEROVERWRITE(LAST)     /* Overwriting of a vol      */ -
          MAXHOLD(100)              /* Number of I/O oper.       */ -
          MAXRETPD(NOLIMIT)         /* Maximum retention         */ -
          MEDIANAME(3480)           /* spec. how to move vols    */ -
          MOVEBY(VOLUME)            /* spec. how to move vols    */ -
          MSG(M)                    /* case for message txt      */ -
          NOTIFY(Y)                 /* Notify volume owners      */ -
          PDA(OFF)                  /* PDA is disabled           */ -
          PDABLKCT(255)             /* number of blocks          */ -
          PDABLKSZ(31)              /* bloksize in K             */ -
          PDALOG(OFF)               /* PDA output disabled       */ -
          PREACS(YES)               /* enable EDGUX100 ACS proc.*/ -
          RETAINBY(VOLUME)          /* spec. how to retain vols  */ -
          RETPD(0005)               /* Default retention         */ -
          REUSEBIN(STARTMOVE)       /* reuse BIN as soon as pos.*/ -
```

```
         SCRATCHPROC(EDGXPROC)       /* ATL/MTL procedure        */ -
         SMFAUD(248)                 /* SMF audit records        */ -
         SMFSEC(249)                 /* SMF security records     */ -
         SMSACS(YES)                 /* enable MV ACS processing */ -
/*       SMSTAPE(UPDATE(EXITS,SCRATCH,COMMAND),PURGE(YES))    ATL*/ -
         SYSID(EGZB)                 /* Name of the system       */ -
         TPRACF(A)                   /* RACF tape support        */ -
         TVEXTPURGE(RELEASE)         /* set an expiration date   */ -
         UNCATALOG(Y)                /* Catalog support          */ -
         VRSCHANGE(VERIFY)           /* No additional action     */ -
         VRSEL(NEW)                  /* New VRS processing       */ -
         VRSJOBNAME(2)               /* DATASETNAME/JOBNAME      */ -
         VRSMIN(0000000100,WARN)     /* Warn if < 100 VRSs       */
/* ****************************************************************** */
MNTMSG   MSGID(IEF233A)      ID(001) VOLUME(016) RACK(016)
MNTMSG   MSGID(IEF233D)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E K')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E R')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEF234E D')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID(IEF455D)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID(IEC501A)      ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E K')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E D')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E R')  ID(001) VOLUME(016) RACK(999)
MNTMSG   MSGID('IEC502E RD') ID(001) VOLUME(017) RACK(999)
MNTMSG   MSGID('IEC502E RK') ID(001) VOLUME(017) RACK(999)
MNTMSG   MSGID(IAT5110)      ID(001) VOLUME(044) RACK(999)
MNTMSG   MSGID(IAT5210)      ID(001) VOLUME(050) RACK(999)
MNTMSG   MSGID(IAT5410)      ID(001) VOLUME(020) RACK(999)
/* ****************************************************************** */
REJECT   ANYUSE(*)
/* ****************************************************************** */
VLPOOL   PREFIX(SC*)                                                 -
         TYPE(S)                                                     -
         RACF(N)                                                     -
         MEDIANAME(3590)                                             -
         EXPDTCHECK(N)                                               -
         DESCRIPTION('3590 DEFAULT SCRATCH POOL')
VLPOOL   PREFIX(N*)                                                  -
         TYPE(R)                                                     -
         RACF(N)                                                     -
         EXPDTCHECK(0)                                               -
         DESCRIPTION('TEST POOL FOR NORBERT')
VLPOOL   PREFIX(F*)                                                  -
         TYPE(R)                                                     -
         RACF(Y)                                                     -
         EXPDTCHECK(Y)                                               -
         DESCRIPTION('FOREIGN CARTRIDGES')
VLPOOL   PREFIX(PP*)                                                 -
```

```
          TYPE(R)                                                  -
          RACF(Y)                                                  -
          EXPDTCHECK(Y)                                            -
          DESCRIPTION('SOFTWARE CARTRIDGES')
/* ************************************************************ */
LOCDEF    LOCATION(SAFE)                                           -
          TYPE(STORAGE)                                            -
          PRIORITY(2000)                                           -
          MEDIANAME(*)                                             -
          MANAGEMENTTYPE(BINS)
LOCDEF    LOCATION(SANJOSE)                                        -
          TYPE(STORAGE)                                            -
          PRIORITY(2000)                                           -
          MEDIANAME(*)                                             -
          MANAGEMENTTYPE(BINS)
LOCDEF    LOCATION(NEWYORK)                                        -
          TYPE(STORAGE)                                            -
          PRIORITY(2000)                                           -
          MEDIANAME(*)                                             -
          MANAGEMENTTYPE(NOBINS)
LOCDEF    LOCATION(IBMATL1)                                        -
          TYPE(LIBRARY)                                            -
          PRIORITY(4800)
/* ************************************************************ */
SECCLS    NUMBER(010)                                              -
          NAME(UCL)                                                -
          SMF(N)                                                   -
          MESSAGE(N)                                               -
          ERASE(N)                                                 -
          DESCIPTION('UNCLASSIFIED')                               -
          MASK('**')
SECCLS    NUMBER(050)                                              -
          NAME(IUO)                                                -
          SMF(Y)                                                   -
          MESSAGE(N)                                               -
          ERASE(N)                                                 -
          DESCIPTION('INTERNAL USE ONLY')                          -
          MASK('**.IUO.**',                                        -
               '**.INTERNAL.**',                                   -
               '**.INT.**')
SECCLS    NUMBER(100)                                              -
          NAME(IC)                                                 -
          SMF(Y)                                                   -
          MESSAGE(Y)                                               -
          ERASE(Y)                                                 -
          DESCIPTION('EXTERNAL USE ONLY')                          -
          MASK(**.IC.**)
```

Archived

You must have completely tested your DFSMSrmm environment before switching to production mode. Carefully review Chapter 8, “Parallel running and validation” on page 309, and be sure to complete all tasks described in that chapter before proceeding.

At the cutover time you have to update the OPTION commands in the EDGRMMxx PARMLIB member definition as shown in Example 9.1.

Change the following options:

- **EXPDTCHECK(Y) to EXPDTCHECK(N)**
  Allow DFSMSrmm the automatic overwriting of unexpired files
- **MOVEMODE(SET) to MOVEMODE(VOLUME)**
  Allow the move of a single volume, if the volume is part of a volume set
- **OPMODE(W) to OPMODE(P)**
  Allow full operation of DFSMSrmm
- **RETAINBY(SET) to RETAINBY(VOLUME)**
  Allow DFSMSrmm to retain any volume individually
- **TPRACF(N) to TPRACF(A)**
  Allow DFSMSrmm to create and maintain RACF TAPEVOL profiles
- **TVEXTPURGE(EXPIRE) to TVEXTPURGE(RELEASE)**
  Allow DFSMSrmm to release a DFSMShsm volume after a call of the EDGTVEXT or EDGDFHSM directly
- **UNCATALOG(N) to UNCATALOG(Y)**
  Allow DFSMSrmm to always uncatalog data sets
- **VRSCHANGE(INFO) to VRSCHANGE(VERIFY)**
  Allow DFSMSrmm to fail the inventory management processing if you have made changes to vital record specifications, but you have not verified the changes.

9.1.1 OPTION command

Following we describe the most important OPTION operands specified:

OPMODE

Specifies the mode of DFSMSrmm

P Protect mode. In protect mode, DFSMSrmm is fully operational. In addition to performing the actions described for manual and record-only mode, now DFSMSrmm rejects tape volumes when it finds an error, instead of giving a message as in WARNING mode.

- Validates all magnetic tape requests and rejects magnetic tape volume mounts under certain conditions. For more information about these conditions, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

MOVEBY(VOLUME)

Specify the MOVEBY operand to move volumes that are retained by DFSMSrmm vital record specifications as a set of volumes or as individual volumes.

VOLUME Specify this value when you want volumes moved as individual volumes.

RETAINBY

Use the RETAINBY option to specify whether DFSMSrmm retains multi-volume sets as a set or as individual volumes.

VOLUME When you retain by volume, DFSMSrmm retains a volume based on vital record specifications and on the volume expiration date. DFSMSrmm does not consider other volumes in the set.

SMSTAPE

Use SMSTAPE to specify how DFSMSrmm updates the TCDB and controls system-managed tape processing.

UPDATE Use UPDATE to select the system-managed tape functions DFSMSrmm provides. The UPDATE operand has three subparameters: EXITS, SCRATCH, and COMMAND. You can specify one or more of the subparameters. When DFSMSrmm is running in PROTECT mode, DFSMSrmm ignores the UPDATE operand and performs processing as if you specified EXITS, SCRATCH, and COMMAND. When you specify a value, DFSMSrmm performs the updates to the TCDB.

- Specify EXITS when you want DFSMSrmm volume status information to override the OAM volume status during entry processing and you want to use the DFSMSrmm VNL exit.
- Specify SCRATCH when you want DFSMSrmm to update the volume status in the TCDB during expiration processing when volumes are returned to scratch status.
- Specify COMMAND when you want to use the RMM TSO subcommands or the DFSMSrmm API to update the TCDB. This controls change of status, TDSI and owner information, eject processing, and manual cartridge entry processing.

PURGE Use PURGE to control how DFSMSrmm affects the TCDB volume records during EJECT processing. The default is PURGE(ASIS) in all operating modes except MANUAL mode. In manual mode, DFSMSrmm provides no support for eject processing.

Archived

YES Specify YES when you want DFSMSrmm to force the TCDB volume records to be purged at eject time.

TPRACF(A)

Specifies the type of RACF tape support for DFSMSrmm to provide. Use this operand when you want DFSMSrmm to maintain the security profiles that protect tape volumes. You can define RACF tape support for pools of volumes within your installation by using the VLPOOL RACF command. The RACF tape support you define can be overridden by the RACF tape support you define with VLPOOL. For more information, refer to “VLPOOL command” on page 75.

A When you specify TPRACF(A), DFSMSrmm creates tape profiles for any volume in your installation, depending on the value of the VLPOOL RACF operand. Refer to “VLPOOL command” on page 75 for details.

For more information on RACF processing and security options, see *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

Note: Before you can switch on the DFSMSrmm RACF tape support, you have to check your existing RACF TAPEVOL profiles and correct them if you have any mismatches.

TVEXTPURGE

Specifies how DFSMSrmm processes volumes are to be purged by callers of EDGTVEXT or EDGDFHSM.

RELEASE DFSMSrmm releases the volume to be purged according to the release actions set for the volume. You must run expiration processing to return a volume to scratch status.

UNCATALOG

Specifies whether or not use DFSMSrmm catalog processing:

Y Specifies that DFSMSrmm should uncatalog data sets when:

- A volume is returned to scratch status, DFSMSrmm uncatalogs all the data sets on the volume.
- The RMM DELETEVOLUME FORCE subcommand is issued for a volume, DFSMSrmm uncatalogs all the data sets on the volume.
- The RMM CHANGEVOLUME DSNAME subcommand is issued for a volume, DFSMSrmm uncatalogs all the data sets on the volume. If the data set name specified on the RMM CV subcommand command matches the data set name on the volume, then DFSMSrmm only uncatalogs subsequent data sets.

Archive

- The RMM DELETEDATASET subcommand is issued for a data set, DFSMSrmm uncatalogs the data set. Also, DFSMSrmm uncatalogs all data sets recorded on the same volume with higher data set sequence numbers.
- A tape data set is overwritten, DFSMSrmm uncatalogs the data set. Also, all data sets recorded on the same volume with higher data set sequence numbers are uncataloged.

During OAM exit processing, DFSMSrmm validates all changes to the TCDB, and fails those that are not allowed. During cartridge entry processing, DFSMSrmm information overrides information provided by OAM. DFSMSrmm always updates the TCDB.

Use SMSTAPE(PURGE) to control the purging or keeping of records in the TCDB during eject processing. Always uncatalog data sets. Use DFSMSrmm catalog processing.

With UNCATALOG(Y), DFSMSrmm can uncatalog data sets even when DFSMSrmm is running in manual, record, or warning mode.

VRSCHANGE

Use VRSCHANGE to specify the action that DFSMSrmm should take during inventory management if you have made any changes to vital record specifications using RMM ADDVRS or RMM DELETEVRS subcommands.

VERIFY Any changes made to vital record specifications must be verified by running EDGHSKP vital record processing using the VERIFY parameter. DFSMSrmm must issue return code zero before EDGHSKP vital record processing can be performed to update the CDS.

9.1.2 VLPOOL command

Set expiration date checking to meet your requirements.

EXPDTCHECK

Specifies to automate the processing of unexpired tape data sets at the pool level:

N Indicates that DFSMSrmm is not to check or validate the expiration date of data sets on a tape. It allows the tape to be overwritten after the system has checked the validity of the tape. DFSMSrmm does not overwrite or replace the expiration date on the tape. This situation happens when a volume is released early, or when someone changes its expiration date or retention period using the RMM TSO subcommands.

Archived

We recommend that you specify N when running a DFSMSrmm managed scratch tape pool. DFSMSrmm lets you reuse tapes only when it is certain the information on them is no longer required. DFSMSrmm can easily automate reuse of tapes if they are still expiration protected, for example, when they are reused as scratch tapes.

Note: When you use the EDGUX100 to tell DFSMSrmm to ignore a volume, DFSMSrmm does not reply to the IEC507D message because the volume is not DFSMSrmm-managed.

9.2 Control access to DFSMSrmm functions

You must ensure that the correct RACF protection will be in place in a production environment to prevent unauthorized access to DFSMSrmm resources. Refer to Appendix C, “Security topics” on page 647.

If you have not already done so, review the RACF and DFSMSrmm definitions in effect, and change them as necessary to meet your production needs.

Be sure to grant RACF access to the FACILITY class resource STGADMIN.EDG.RESET.SSI that prevents users from disabling the DFSMSrmm subsystem interface. Once in production, you do not want the OPT=RESET option to be used.

9.3 Activate exits

During the conversion phase, you merged some exits to allow DFSMSrmm to run in parallel with TLMS, and test that the exits work correctly in an DFSMSrmm environment without affecting the production operations of TLMS.

Now you can remove these merged exits using the DFSMSrmm-supplied version, because in a production environment, you will no longer be using TLMS.

If the code of one or all of the following exits has been merged to allow the use of both tape management products, you must consider removing the merged version of the exit and using the DFSMSrmm version when running in production:

ARCTVEXT DFSMShsm tape volume exit (only in DFSMSrmm prior to Version 1, Release 4)

IGXMSGEX DFSMSrmm uses the DFSMSdfp MSGDISP exit to update tape drive displays

CBRUXENT	OAM cartridge entry exit
CBRUXEJC	OAM cartridge eject exit
CBRUXCUA	OAM cartridge change use attributes exit
CBRUXVNL	OAM volume not in library exit

9.4 Remove TLMS

In this section we describe how you can remove or deactivate your current tape management system without a need of an MVS restart (IPL). The steps you have to perform are:

- Deactivate program interfaces from the operating system.
- Stop TLMS.
- Remove the code from the system libraries.

If your TLMS installation changed some of the IBM modules, you must remove those changes and restore the IBM modules to the latest PTF level before the changes were made.

Use either SMP/E RESTORE processing or the AMASPZAP utility to remove those changes. You may need to IPL with the CLPA option to activate the updated code. These modifications can be removed without an IPL if TLMS uses dynamic updates to IBM modules. Refer to the installation manual of TLMS if you are not sure which method to use.

Once you have removed the interfaces, you can stop tape activity for TLMS, and delete the libraries from the system.

Important note: You can stop TLMS only when you are ready to start DFSMSrmm in production. Stopping TLMS must be done after stopping all tape activity, in a single step. Refer to 9.8, “Starting DFSMSrmm” on page 375 for the correct timing of this activity.

9.5 Modifying CA-Disk

You should modify your CA-Disk (formerly SAMS:Disk) to support only the DFSMSrmm tape management system. To do this, you have to update the CA-Disk SYSPARMs.

Change the `TMSCTLEX` from `ADSTH014` to `ADSTH016` value, and delete the `TAPCTLEX` entry from your CA-Disk PARMLIB.

For more information, refer to the CA-Disk product publications.

9.6 Inventory management

During the parallel running and validation phase, you should have properly customized your inventory management procedures to allow both DFSMSrmm and the other vendor product to update databases on a daily basis and perform retention and movement processing.

In the previous phase, you produced retention and movement reports from both tape management products. You no longer have a need for the vendor product reports, and now you use only the equivalent DFSMSrmm reports.

Because you no longer need to run housekeeping jobs on TLMS, if you have a scheduling management system like the Tivoli Workload Scheduler or a similar product, you can remove the unwanted jobs or steps from the schedule plan.

In addition to inventory management jobs, there are other reports that you no longer need, such as retention and movement reports that you previously produced on a regular or periodic basis. Substitute these reports with the equivalent DFSMSrmm reports, which now will be the only reports you need to produce.

Be sure that you have removed all of the vendor's product jobs from the schedule plan, because they can no longer run once the code of TLMS has been removed from the system.

If you have a scheduling management product, you probably changed the daily job scheduling plan to allow both DFSMSrmm and the vendor product to update their databases simultaneously when running in parallel. Now you must delete all references to the jobs of the vendor product, because once in production, these jobs will no longer run.

Production controllers should be aware of these changes to ensure that they will not impact the production scheduling of other jobs.

Important: You must pay attention to possible interfaces between TLMS and other products; for example, CA-1, the restart and rerun package from Computer Associates.

DFSMSrmm uses the special ABEND VRS retention policy to control tape data sets generated by mistake as result of an abend. You can achieve the same function in the DFSMSrmm environment.

Catalog resynchronization (optional)

At this point you should check that all your catalogs are in sync with your DFSMSrmm control data set. If there is a doubt that all catalogs are in synchronization or if you have not switched on catalog synchronization, run the catalog synchronization as described in detail in 8.1.10, “Catalog synchronization” on page 329.

Note: Rerun the catalog synchronization if you are manually adding volumes in MASTER or USER status with data set information or you are changing your Catalog environment (MERGECAT, IMPORT, EXPORT).

Tip: We recommend you enable this function to improve the performance of the daily vital record and expiration processing.

9.7 Interface to tape libraries

If you have an automated tape library installed in your production system, the only way you can test it before going into production with DFSMSrmm is to run DFSMSrmm in record or warning mode, and prevent DFSMSrmm from updating the TCDB catalog.

You can do this by activating only the interface provided with TLMS, and not running EDGHSKP EXPROC processing, or by running DFSMSrmm in record mode. In record mode DFSMSrmm makes no changes to the TCDB. When you switch DFSMSrmm to production, you now use DFSMSrmm data as input instead of the data from your existing product. Use the SMSTAPE operand in the EDGRMMnn PARMLIB member to prevent TCDB updates in other OPMODES.

9.7.1 System-managed IBM 3494 and IBM 3495 tape libraries

You must use the DFSMSrmm version of the following exits:

CBRUXENT OAM cartridge entry
CBRUXCUA OAM cartridge change use attribute
CBRUXEJC OAM cartridge eject
CBRUXVNL OAM cartridge volume not in library

If you previously changed one or more of these exits with a merged version to allow DFSMSrmm and the other product to work together, you must now substitute them with the DFSMSrmm version.

If you have not yet tested your tape library with DFSMSrmm, you must test it in production to ensure that everything works correctly. Refer to 9.9, “Final testing” on page 377 for more information.

If you are moving cartridges from an IBM ATL to an outside storage location or to your SHELF location, you can use the sample REXX exec shown in Example 9-2 to check that all cartridges ejected sucessfully.

Example 9-2 REXX exec to check cartridge ejects

```
/******************* REXX ***********************************/
/*                                                           */
/*   Check for waiting ejects                                */
/*                                                           */
/*************************************************************/
TRACE Normal
PARSE UPPER ARG oam_id .
IF oam_id = '' Then oam_id = 'OAM'
error=false
aejects = 0
wejects = 0
Do until remainingejects = 0 | error=true
   ADDRESS "TSO" "CONSPROF SOLDISPLAY(YES)"
   cc = rc
   ADDRESS "TSO" "CONSOLE ACT"
   cc = rc
   ADDRESS  "CONSOLE" "F "oam_id",Q,A,SUMMARY"
   cc = rc
   mrc=GETMSG(line.,SOL,,,10)
   IF cc=0 & mrc=0 Then aejects=word(line.4,2)
   else error=true
   ADDRESS  "CONSOLE" "F "oam_id",Q,W,SUMMARY"
   cc = rc
   mrc=GETMSG(line.,SOL,,,10)
   IF cc=0 & mrc=0 Then wejects=word(line.4,2)
   else error=true
   remainingejects = wejects+aejects
   IF remainingejects > 0 Then
     do
       say 'Waiting for 'total' ejects'
       mrc=GETMSG(line.,SOL,,,60)  /* wait for 1 minute */
     end
   Else
       say 'No ejects remain to be actioned'
   ADDRESS "TSO" "CONSOLE DEACT"
end
exit(cc+mrc)
```

Archived

9.7.2 Other tape libraries

Whatever tape library is installed in your system, DFSMSrmm can support it through a user interface that tells the library which volumes to return to scratch status and which volumes must be ejected and moved to another storage location.

With DFSMSrmm's reporting and command capabilities it is simple to build this interface, using REXX, DFSORT, DFSMSrmm utilities, and TSO RMM subcommands.

You must analyze what the vendor tape library requires as input for retention and movement processing and then customize DFSMSrmm to provide the library with the correct input.

Be sure to keep the vendor tape library catalog and the DFSMSrmm database synchronized to avoid any mismatch that would cause rejection of tape mounts.

STK tape libraries

Below we provide you with a simple interface example between an STK Silo tape library and DFSMSrmm to eject tapes that need to be moved to another location.

After running DFSMSrmm inventory management (storage location management processing DSTORE), you can create a list of all volumes that are moving by using DFSMSrmm commands. Follow these steps:

1. Use DFSMSrmm commands as shown in Example 9-3.

Example 9-3 RMM SEARCHVOLUME subcommands with specific destinations

```
RMM SV VOLUME(*) DESTINATION(LOCAL) CLIST(' ',' -')
RMM SV VOLUME(*) DESTINATION(REMOTE) CLIST(' ',' -')
RMM SV VOLUME(*) DESTINATION(DISTANT) CLIST(' ',' -')
RMM SV VOLUME(*) DESTINATION(my_loc) CLIST(' ',' -')
```

Note: Unless you exploit the capability of APAR OW30790, and use a pre-allocated RMMCLIST DD name, you can use only one TSO RMM subcommand at a time, because each call to the CLIST option will overwrite the data created before.

The resulting list, which is stored in the RMMCLIST data set, or the data set *prefix*.EXEC.RMM.CLIST consists of VOLSERs only. The data set uses blocked, variable-length records, with a logical record length of 255. Where *prefix* is the TSO user PROFILE PREFIX that the user has currently defined. If you have specified *NOPREFIX*, DFSMSrmm allocates a data set using the RACF user ID or the jobname, to avoid using EXEC.RMM.CLIST.

If you are using the TSO RMM subcommands in a batch job, you can set the prefix in front of the TSO RMM subcommand. Example 9-4 shows sample JCL.

Example 9-4 Using TSO RMM subcommands in a batch JOB

```
//RMMMOVE JOB  ....
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 PROF PREFIX(prefix)
 RMM SV VOLUME(*) DESTINATION(my_loc) CLIST(' ',' -')
/*
```

2. Convert the list from variable-length records to fixed-length records. HSC/MVS control statement syntax conventions: The control statement for each utility program consists of a command (indicating the utility function) followed by parameters, as applicable, in 80-character card-image records (see the *HSC/MVS Programmer's Guide, Volume 2* provided with the product).
3. Edit this list and create an EJECT statement as shown in Example 9-5, which can be used as input for the HSC utility.

Example 9-5 HSC EJECT statements

```
EJECT VOLSER( -
volser1 -
volser2 -
volser3 -
. . . . . -
) CAP(cap-list)
```

Please be aware that each single EJECT command causes an operator interaction.

4. Run the HSC update utility, SLUADMIN (see the *HSC/MVS System Programmer's Guide, Volume 1* provided with the product).

The HSC initialize cartridge utility (SLUADMIN)

Example 9-6 shows sample JCL that you can use to initialize cartridges with the HSC SLUADMIN utility by using DFSMSrmm EDGINERS processing.

Example 9-6 Sample EDGINERS JCL

```
//JOBTINIT JOB (ACCOUNT),PROGRAMER
//*
//TINIT    EXEC PGM=SLUADMIN,TIME=1440,PARM=MIXED
//TAPE     DD   UNIT=(580,1,DEFER),DISP=NEW
```

Archived

```
//*                              EDGINERS TAPE UNIT DD
//*
//SLSTAPE  DD   UNIT=AFF=TAPE            INIT CARTRIDGES INIT TAPE
//*                                      UNIT DD.
//SYSIN    DD   UNIT=VIO,SPACE=(TRK,1)   EDGINERS CONTROL DATASET
//SLSPRINT DD   SYSOUT=*                 UTILITY MESSAGES
//SYSPRINT DD   SYSOUT=*                 EDGINERS MESSAGES
//SLSINIT  DD   *
 INIT LABEL(SL) VOL(******)
/*
//SLSIN    DD   *
//*
* WHEN USING THE DFSMSrmm TAPE INITIALIZATION PROGRAM EDGINERS, YOU
* MUST SPECIFY THE 'PROGRAM' PARAMETER ON THE INIT CARTRIDGES
* CONTROL STATEMENT.
 INITIALIZE CAP(000) PROGRAM(EDGINERS) OPTION(SCRATCH)
/*
```

The HSC scratch conversion SLUCONDB utility

The HSC scratch conversion SLUCONDB program is provided to generate scratch transactions from DFSMSrmm. SLUCONDB calls the routine SLUDRRMM to read the DFSMSrmm report extract file if RMM is specified in the EXEC JCL statement. This program utilizes the DFSMSrmm EDGRVEXT macro to map the layout of the extract file. The following list shows the fields that are used:

- RVTYPE
- RVVOLSER
- RVLCDATE
- RVNAME
- RVSTATUS
- RVEXPDT
- RVEXPDTO
- RVLABEL

SLUDRRMM processes the extract and passes the volume record information to SLUCONDB, which builds scratch card images to input to SLUADMIN. Volume status scratch (scratch or non-scratch) is then updated in the HSC CDS for each volume record in the extract file.

Dates on the DFSMSrmm report must be in Julian date format (EDGHSKP run with PARM 'DATEFORM(J)'). Tapes listed on the DFSMSrmm report without expiration dates will be skipped by SLUDRRMM. Example 9-7 is an example of the DFSMSrmm and HSC synchronization JCL.

Example 9-7 DFSMSrmm and HSC synchronization JCL

```
//JOBSCUP  JOB  (ACCOUNT),PROGRAMMER
//S1       EXEC PGM=EDGHSKP,PARM='RPTEXT,DATEFORM(J)'
//MESSAGE  DD  DSN=RMM.MESSAGE.DATASET,DISP=SHR
//REPTEXT  DD  DSN=RMM.REPORT.EXTRACT,DISP=SHR
//S2       EXEC PGM=SLUCONDB,PARM=('RMM,SCRPOOL(SL)',
         '87100-87110,MIXED')
//SLSTMS   DD  DSN=RMM.REPORT.EXTRACT,DISP=SHR
//SLSSOUT  DD  DSN=&&SCUPINPT,DISP=(NEW,PASS),UNIT=,SPACE=DCB=
//SLSPRINT DD  SYSOUT=A
//S3       EXEC PGM=SLUADMIN
//SLSIN    DD  DSN=&&SCUPINPT,DISP=(OLD,DELETE)
//SLSPRINT DD  SYSOUT=A
//
```

The SLUCONDB utility, when PARM=RMM is specified, checks in the DFSMSrmm extract filter to see if the MEDIANAME of the tapes being returned to scratch is either 3480 or 3490. If you are using a different MEDIANAME, for example CARTS, and it is not found, no records are selected by this routine and the inventory of the robot is not synchronized with DFSMSrmm. There are two possibilities for solving this problem:

1. Update the MEDIANAME used in the STK SLUADMIN program by updating the SLUDRRMM source module. After that, compile and link edit it.

   Example 9-8 shows you the original program code.

Example 9-8 Original SLUDRRMM macro

```
......
READRCD  EQU   *
         LA    R2,BUFFER2+4              LOAD ADRESS OF INPUT BUFFER
         GET   INPUT,BUFFER2             READ RECORD
         USING RVEXT,R2                  ESTABLISH ADRESSABILITY TO RCD.
         CLI   RVTYPE,C'V'               WAS IT A VOLUME RECORD?
         BNE   READRCD                   NO - READ NEXT RECORD.
         CLC   RVNAME,=CL8'3480'         WAS RECORD FOR A 3480 VOL?
         BE    READCONT                  YES - CHECK STATUS.
         CLC   RVNAME,=CL8'3490'         WAS RECORD FOR A 3490 VOL?
         BNE   READRCD                   NO - READ NEXT RECORD.
READCONT CLC   RVSTATUS,=CL8'STATUS'     WAS RECORD IN SCRATCH STATUS?
         BNE   READRCD                   NO - READ NEXT RECORD.
.......
```

And the modified program is shown in Example 9-9.

Example 9-9 Updating SLUDRRMM macro

```
.......
READRCD  EQU   *
```

```
         LA    R2,BUFFER2+4             LOAD ADRESS OF INPUT BUFFER
         GET   INPUT,BUFFER2            READ RECORD
         USING RVEXT,R2                 ESTABLISH ADRESSABILITY TO RCD.
         CLI   RVTYPE,C'V'              WAS IT A VOLUME RECORD?
         BNE   READRCD                  NO - READ NEXT RECORD.
         CLC   RVNAME,=CL8'3480'        WAS RECORD FOR A 3480 VOL?
         BE    READCONT                 YES - CHECK STATUS.
         CLC   RVNAME,=CL8'CARTS'       WAS RECORD FOR CARTRIDGES?
         BE    READCONT                 YES - CHECK STATUS.
         CLC   RVNAME,=CL8'3490'        WAS RECORD FOR A 3490 VOL?
         BE    READCONT                 YES - CHECK STATUS.
         CLC   RVNAME,=CL8'3590'        WAS RECORD FOR A 3590 VOL?
         BNE   READRCD                  NO - READ NEXT RECORD.
READCONT CLC   RVSTATUS,=CL8'STATUS'     WAS RECORD IN SCRATCH STATUS?
         BNE   READRCD                  NO - READ NEXT RECORD.
.......
```

Note: You should define in the routine all the MEDIANAMES that you have specified in the VLPOOL command. In this example, you define a MEDIANAME of 3480, 3490, 3590, or CARTS.

2. Leave the routine as it is now, and specify 3480 or 3490 in the MEDIANAME of the VLPOOL command.

STK VSM libraries

If you have installed a STK VSM following the STK recommendations:

Protect MVC volumes through RACF:

1. Define a profile in the TAPEVOL class for Multi-Volume Cartridge (MVC) as shown Example 9-10.

Example 9-10 Add a TAPEVOL profile

```
RDEFINE TAPEVOL CVC024 UACC(NONE)
```

2. Allows user ID VSM8HSC (STC user ID) update access to MVC CVC024 as shown in Example 9-11.

Example 9-11 Permit HSC to MVC's

```
PERMIT CVC024 CLASS(TAPEVOL) ACCESS(UPDATE) ID(VSM8HSC)
```

3. Allow STC the use of LISTVOL and CHANGEVOL commands. See *z/OS Security Server RACF Command Language Reference,* SA22-7687 for more information about all the resources defined in the RACF FACILITY class.

Archive

Allows user ID VSM8HSC (STC user ID) control access as shown in Example 9-12.

Example 9-12 Permit HSC to RMM resources

```
PERMIT STGADMIN.EDG.MASTER ACCESS(CONTROL) ID(VSM8HSC)
```

4. Protect VTCS commands
   Define a profile in the OPERCMDS class for all VTCS commands. You can use a RACF RDEFINE OPERCMDS as shown in Example 9-13.

Example 9-13 ADD an OPERCMDS resource

```
RDEFINE OPERCMDS subsystemname.VT UACC(NONE)
```

5. Allow user SAM15 update access to all VTCS commands you have previously defined as shown in Example 9-14.

Example 9-14 PERMIT VSM to OPERCMDS

```
PERMIT subsystemname.VT CLASS(OPERCMDS) ID(SAM15) ACC(UPDATE)
```

Caution: VTCS has an automatic interface to notify DFSMSrmm when a VTV becomes scratch, but DFSMSrmm does not notify VTCS when unscratching a VTV by changing the volume status to non-scratch. If this is done, you must also run the HSC SLUADMIN utility to unscratch the VTV for VTCS. Otherwise, you may encounter a mount failure when VTCS attempts to service a scratch mount request.

Memorex tape libraries

Memorex has announced the availability of LMSrmm, which provides integration of the DFSMSrmm tape database with LMS automation tape library control functions.

LMSrmm provides DFSMSrmm support for all models of the Memorex/Telex 5100, 5200, and 5400 ATL product family.

Refer to the Memorex documentation on how to install this product. You will have to install some LMSrmm code and modify the EDGUX100 DFSMSrmm exit.

9.8 Starting DFSMSrmm

You may need to build a final version of the converted DFSMSrmm CDS to ensure that the production environment is in sync with the vendor product; at this point rerun the conversion process.

To obtain a final updated DFSMSrmm database, you may have to rerun the entire conversion process.

By now, you should have a tailored jobstream that can be run very quickly to reconvert the latest data. By using a prepared and tested jobstream, you can reduce to a minimum the elapsed time during which tape activity must be interrupted. The jobstream uses as input the vendor database from the point of time where you stopped tape activity, and gives you a correctly converted DFSMSrmm CDS ready to be used in production.

Once you have obtained the final version of the DFSMSrmm CDS, be sure that you have performed all steps previously described in this chapter and detailed in Table 9-1.

Table 9-1 Steps required to go into production with DFSMSrmm

1	Review and change DFSMSrmm PARMLIB options
2	Review and change DFSMShsm and other exits
3	Remove vendor product's IBM module modifications
4	Modify or remove inventory management jobs
5	Modify or remove tape management reports
6	Install or modify exits
7	Set up other vendor robots to work with DFSMSrmm
8	Review and modify automatic job scheduling plan
9	Stop tape activities
10	Final reconvert to DFSMSrmm from vendor product
11	Do an IPL if necessary
12	Start DFSMSrmm in production in PROTECT mode

You must ensure that the people who will be using DFSMSrmm in production have sufficient skills to correctly operate the tape management system in the new environment. The education sessions that you scheduled during the conversion phases should have included these people.

Archived

9.9 Final testing

You are now in a production environment with DFSMSrmm started in protect mode. The most important thing to do now is to retest the vital functions of DFSMSrmm and test other functions and interfaces that you could not test before going into production, such as:

- DB2® logging and backup
- RACF tape protection
- Robot interface
- Non-system managed libraries SMS ACS support

The first simple test that you can do to verify DFSMSrmm functions in protect mode is to mount a wrong volume and see whether DFSMSrmm rejects it. Try to mount a specific master volume as input to a scratch request, or a volume with a different label when asking for a specific mount request. The volume mounted for the first test must be rejected by DFSMSrmm with the message shown in Figure 9-1.

```
EDG4027I VOLUME volser REJECTED. IT IS NOT A SCRATCH VOLUME
         AND MOUNT REQUEST WAS NON-SPECIFIC
```

Figure 9-1 EDG4027I message

For the second test, it must be rejected with the message shown in Figure 9-2.

```
EDG4050I VOLUME volser REJECTED. IT IS NOT EQUAL
         TO THE VOLUME REQUESTED
```

Figure 9-2 EDG4050I message

With these simple tests, you can see whether DFSMSrmm will reject any wrong tape mount instead of issuing a warning message as it did while running in warning mode.

9.9.1 DB2 logging and backup

DB2 uses its own retention process for archive logs and image copy of tables and table spaces.

If you want to control DB2 tape activities with VRS policies, you must change the default retention period in the DB2 installation options to a date closer than 99365, which in DFSMSrmm means *never expire*. With this change, you can protect DB2 archive and image copy tape data sets with VRS, masking the data set name as desired.

Archive

You must synchronize the DB2 and DFSMSrmm expiration dates to avoid a mismatch between the DB2 system tables and DFSMSrmm CDS information. You can use a VRS definition with UNTILEXPIRED for DB2 archive log files to synchronize the retention period specified in the DSNZPARM DB2 initialization module with DFSMSrmm retention.

You can test the above VRS rules by running DB2 jobs that produce archive logs and image copy data sets on tape. Then, after running DFSMSrmm housekeeping, use the following RMM command shown in Figure 9-15.

Example 9-15 RMM LISTVOLUME subcommand

```
RMM LV volser ALL
```

and check whether the AVAILABILITY field is "VITAL RECORD" and the RETENTION DATE is as desired.

9.9.2 RACF tape protection

The DFSMSrmm-RACF interface automatically handles the definition of RACF profiles using either the TPRACF(A) or TPRACF(P) OPTION PARMLIB control statement.

When you activate either of these options, you must test using RACF and DFSMSrmm to verify the correct working of this interface. You can submit test jobs that use a new tape data set name and then check whether the RACF profile has been created, using commands or RACF ISPF panels.

When you delete these data sets using DFSMSrmm commands, any discrete RACF data set profiles should be deleted automatically. Discrete TAPEVOL profiles are managed by DFSMSrmm as data is created and at return to scratch time.

Be sure to check RACF tape profiles for TAPEVOL class, and also DATASET class if TAPEDSN is used.

9.9.3 Robot interface

If you have an IBM tape library, you can test the DFSMSrmm interface simply by releasing a volume from DFSMSrmm and then verifying that the TCDB record reflects the DFSMSrmm CDS information after running the housekeeping process.

Archived

Use the command in Example 9-16 for another test you can do. The location should be different than the actual location name. You can use the location SHELF for example.

Example 9-16 RMM CHANGVOLUME subcommand to change the location

```
RMM CV volser LOCATION(loc_name)
```

This will cause the specified volume to be ejected from the tape library with the OAM message shown in Figure 9-3.

```
CBR3010I Volume volser ejected from library library-name.
         Place in shelf location shelfloc.
```

Figure 9-3 CBR3010I message

The TCDB record will be deleted. It will be rebuilt once the volume is inserted into the library using the information from the DFSMSrmm CDS.

If you are using another vendor's tape library, you must test the interface used with the library for retention and movement processing. Try using jobs or commands that cause ejection or release of volumes. Use the DFSMSrmm status information to drive input to the software that controls the robot tape library.

9.9.4 Non-system managed libraries SMS ACS support

If you have enabled the DFSMSrmm SMS ACS as described in Chapter 4., “DFSMSrmm ACS support” on page 137 to support scratch pooling and VRS policy management using SMS ACS management class and storage group routines, we recommend you to do some final testing before you cut over to production. If you have implemented scratch pooling in your storage group ACS routine, you might run a new allocation as shown in Example 9-17, and check that DFSMSrmm picks up the right pool name.

Example 9-17 Sample JCL for non-specific tape volume mount

```
//D109123K JOB ,'RMM',TIME=NOLIMIT,
//         MSGLEVEL=(1,1),MSGCLASS=H,CLASS=A
//IEBGENER EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT2   DD DISP=(,KEEP,DELETE),
//         DSN=TAPE.MVSTAPE.MNT06.MODS1G01.TAB011HJ,
//         DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,DIAGNS=TRACE),
//         UNIT=(3480,,DEFER),LABEL=(1,SL)
```

```
//SYSUT1   DD *
SAMPLE DATA
```

The following checklist might help you:

- Is the PARMLIB SMSACS(YES) enabled?
- Are the management classes and storage groups defined to SMS?
- Have you assigned valid storage group names to the VLPOOL definitions you plan to use for scratch pooling?
- Do you check for the ACS routine environment variable &ACSENVIR 'RMMPOOL' and 'RMMVRS' in your management class and in your storage group ACS routine?

9.10 Commitment to DFSMSrmm

If the final testing of DFSMSrmm functions is successful, you can be sure that your conversion is complete. If, however, something goes wrong during the final testing phase, you have two options:

- Go back to the parallel running phase.
- Fall back to another vendor product and plan reconversion when ready.

The option you select will depend on how serious the problem is that prevents DFSMSrmm from working correctly in a production environment.

Remember that in both situations you must use an old version of the vendor's product database, which might not be updated to reflect the tape activities since you started DFSMSrmm in production to remove TLMS.

You can avoid this situation by testing your production environment on a test system with DFSMSrmm in protect mode and the vendor product removed. You can do this early in the conversion process to anticipate and plan around any problems detected.

If some VRS definitions are wrong, and you did not correct them during the parallel running and validation phase, you might have tapes that are assigned to SCRATCH even though they contain valid data for your production system.

Remember that when running DFSMSrmm in *protect* mode, you cannot read a SCRATCH or PENDING RELEASE tape, so you need a method to read the contents of those tapes for recovery. To recover without fall back, you can use the DFSMSrmm command shown in Example 9-18 if the volume is in SCRATCH status.

Example 9-18 RMM CHANGEVOLUME subcommand to change the volume status

```
RMM CV volser STATUS(USER)
```

Use the command in Example 9-19 if the volume is in PENDING RELEASE status.

Example 9-19 RMM CHANGEVOLUME subcommand to change the retention

```
RMM CV volser RETPD(1)
```

After you have issued these commands you can read either kind of tape.

Using DFSMSrmm commands, you can solve other mismatches in VRSs without the need to return to the other vendor's product. For example, if you have a wrong location name for a set of tapes, you can change the location for those tapes, and confirm the movement to DFSMSrmm by issuing the command shown in Example 9-20.

Example 9-20 RMM CHANGEVOLUME subcommand to confirm the move

```
RMM CV volser LOC(new_loc) CMOVE
```

This command will have the volume recorded as being in the *new_loc* location. Without the CMOVE operand, only the destination is set.

If you have many volumes, you can use the SEARCHVOLUME command and create executable commands by using the CLIST subparameter as shown in Example 9-21.

Example 9-21 RMM CHANGEVOLUME subcommand to move a volume

```
RMM SV VOL(*) OWN(*) LIMIT(*) LOCATION(wrong_loc)      -
       CLIST('RMM CV ',' DEST(new_loc) CMOVE')
```

Then execute the CLIST generated. You can use either the TSO RMM subcommand as shown in Example 9-22.

Example 9-22 RMM SEARCHVOLUME with LOCATION subparameter

```
RMM SV VOL(*) LOC(new_loc)
```

or use EDGRPTD to generate an inventory report.

Archived

If you cannot recover the error situation in your production environment, and you have to fall back to the previous environment, you must follow the steps detailed in Table 9-2.

Table 9-2 Fallback plan

1	Stop DFSMSrmm and stop tape activities.
2	Reinstall vendor product code.
3	Restore the last version of vendor product's database.
4	Add the last tape activities not recorded, if possible.
5	Restore changes made after parallel running (DB2, RACF, robots).
6	Remove or modify exits.
7	Start TLMS.
8	Start DFSMSrmm in record or warning mode.
9	Solve the problem that caused the fallback.
10	Plan another cutover to production.

Remember that timing is the critical factor in deciding whether or not to fall back. For example, if you have had DFSMSrmm in production for five working days, it is difficult to fall back. If you have had DFSMSrmm in production for one hour, the fallback is quite easy.

You need to be aware of the tape activity in your installation. We strongly recommend that you consider looking at a fallback procedure in your planning process, regardless of whether you end up executing the fallback. You need to identify the point where you can back out and develop the fallback plan based on that point. Unfortunately, we are not able to recommend a specific point, because it varies from installation to installation.

If TLMS can coexist with DFSMSrmm running in protect mode, we recommend that you continue to use it. We suggest that you test the process first. If you conclude that running DFSMSrmm and TLMS in parallel is not going to affect your operating environment, you should proceed with such an approach.

Because DFSMSrmm can operate in several modes, we strongly recommend that you run the conversion process as many times as you like until you are comfortable with the result, before taking DFSMSrmm into production. This approach should help eliminate any concern for fallback from DFSMSrmm, but it does not necessarily mean that you do not have to plan the fallback.

9.11 Final activities

After your successful DFSMSrmm cutover to production, you may consider performing some final activities, such as:

- Uninstall and delete TLMS completely
- Cleaning up your DFSMSrmm implementation
- Implement additional functions

9.11.1 Uninstalling TLMS

For a complete removal of TLMS, you should perform the following activities:

- Remove the *patches*.
- Delete the procedure(s).
- Delete the JCL.
- Delete the old data sets.
- Delete the RACF profiles.
- Delete the old programs.

9.11.2 Cleaning up your DFSMSrmm implementation

Cleaning up the DFSMSrmm implementation consists of:

1. Reducing VRS policies through exploitation of data set name masks
2. Centralizing control of retention and movement through VRS
3. Deleting volumes that no longer exist but were included in the conversion

Reduce VRS policies

Vendor tape management systems have only recently started to provide generic or filter capability for defining data set names for retention and movement. DFSMSrmm has always included this capability. During conversion, most of the retention and movement policies being converted use one of the following:

- A fully qualified data set name
- A partially qualified data set name
- A data set name index

By using data set name masks as filters you can significantly reduce the number of VRS policies required as long as there is some commonality to the data set names used. For example, the single mask:

```
DAILY.BACKUP.V*
```

is equivalent to all of the data set names:

```
DAILY.BACKUP.VVOL001
DAILY.BACKUP.VVOL002
```

Archived

```
DAILY.BACKUP.VVOL003
DAILY.BACKUP.VVOL004
DAILY.BACKUP.VVOL005
```

This is a very simple case, but through the same principles and use of the %, *, ~, and **generic characters, you can build masks in many ways.

You can only manage data sets through a common data set name filter if the data sets have identical retention and movement needs.

When you define a new VRS to DFSMSrmm, and the data set name mask is less specific than the mask that is already defined to DFSMSrmm, it is not used. So, as part of the consolidation you can define VRSs with data set names without deleting the existing VRSs. You can then run DFSMSrmm inventory management VRSEL processing and check the retention report to see whether any of the new VRSs are used. If any of the new VRSs are used, you are bringing additional data sets under VRS control. This may not be desirable, so you need to review the retention report carefully to see whether the correct results are obtained. It might be that these data sets should have been retained, but through an oversight in the past, they were not.

Once you are satisfied with the report, you can clean up the old VRSs by deleting them from DFSMSrmm. Rerun the DFSMSrmm inventory management VRSEL processing to generate a report that shows which data sets are being retained by which VRS. The report should now show that the data sets covered by the VRSs you deleted are covered by the new, generic VRSs you defined.

Having generic VRSs that cover many different data sets with the same retention requirements makes it simpler to modify the retention requirements. There are fewer changes to make.

VRS cleanup after conversion

When you implement the VRS cleanup after conversion, you define new VRSs and then delete the old VRSs:

1. Analyze your current VRSs. Consider the data set names and look for those with common naming conventions and identical retention and movement needs.
2. Build a list of new data set names and the retention criteria they need.
3. Use the ADDVRS subcommand to define the new VRSs.
4. Run EDGHSKP with PARM=VRSEL and the REPORT DD statement included. Use the report output to ensure that the new VRSs are not yet being used for any existing data sets.
5. Use the DELETEVRS subcommand to remove the policies that are no longer required.

6. Run EDGHSKP with PARM=VRSEL and the REPORT DD statement included. Use the report output to ensure that the new VRSs are now being used for the data sets previously retained by the deleted VRSs.

VRS cleanup as part of conversion

When you do the VRS cleanup as part of the conversion process, you do not have VRSs to delete. You have either the K-Records that are input to EDGCNVT or the ADDVRS commands generated by EDGCNVT. We recommend that you do the cleanup on the ADDVRS commands, by editing them to add data set names and removing the redundant names. When you load the DFSMSrmm CDS, you do not use the records that EDGCNVT writes to the VRSLIST output file; you execute the modified ADDVRS commands once DFSMSrmm is started with the new CDS. The VRS cleanup consists of the following steps:

1. Run EDGCNVT; validate that converted retention polices meet your needs.
2. Review the ADDVRS commands in the VRSCMDS file created by EDGCNVT. Consider the data set names and look for those with common naming conventions and identical retention and movement needs.
3. Build a list of new data set names and the retention criteria they need.
4. Modify the ADDVRS subcommands to include the new VRSs and remove the VRSs that are not required.
5. Modify the LOADRMM job to remove the VRSLIST file created by EDGCNVT. After you start DFSMSrmm with the new CDS, you execute the ADDVRS commands you have prepared.
6. Run EDGHSKP with PARM=VRSEL and the REPORT DD statement included. Use the report output to ensure that the new VRSs are now being used to provide the retention you need. Each time that you reconvert, use the prepared ADDVRS commands, not the VRSLIST that EDGCNVT generates.

Centralize control of retention and movement

Under TLMS retention criteria could come from several places:

- JCL RETPD or EXPDT keywords
- JCL-specified keyword date, such as 99000
- SMS data class
- A default retention
- A retention policy

DFSMSrmm also provides support for the above ways of specifying retention. The first three ways of specifying retention allow the user to control retention, and the last two ways provide a way for the installation to control retention. Without a centrally controlled policy, and without enforcement of that policy, you have no way of controlling the retention of data on tape.

With DFSMSrmm you can enforce central control over all retention if you want. Use the VRSs to define all retention needs. Use the RETPD and MAXRETPD values in the DFSMSrmm PARMLIB to set a default retention period and prevent the volume expiration date from overriding the retention information in the VRS.

Specifying MAXRETPD(0) forces the expiration date of all volumes to be set to the date they are assigned from scratch status. MAXRETPD cannot be less than the RETPD value, so RETPD(0) also must be specified. You may decide to specify just a small value so that new data is kept for a minimum time period, for example, MAXRETPD(2) RETPD(2).

Note: Applications such as the DFSMShsm that use a permanent retention date like 99365 to retain data are affected by reducing the MAXRETPD value. DFSMSrmm no longer retains them forever. You must ensure that VRSs are defined that cause the data sets they create to be retained until the application releases them.

You can define a default retention period for all data sets using VRS with the command shown in Example 9-23.

Example 9-23 RMM subcommand ADDVRS DSN(**)

```
RMM ADDVRS DSN('**') WHILECATALOG
```

The previous VRS covers all data sets not covered by any other VRS. To prevent the VRS management value from being used, you can define a default VRS as shown in Example 9-24.

Example 9-24 RMM subcommand ADDVRS DSN(.**)*

```
RMM ADDVRS DSN('*.**') WHILECATALOG
```

To implement centralized control of retention, follow these steps:

1. Define a default VRS as in the previous sample.
2. Change the DFSMSrmm PARMLIB members MAXRETP and RETPD to be either 0 or a small value.
3. Restart DFSMSrmm with the new PARMLIB member, using the operator command shown in Example 9-25, assuming the member name is EDGRMM01.

Example 9-25 Modifying DFSMSrmm

```
F DFRMM,M=01
```

4. For existing volumes that have an expiration date that has not yet been reached, change the volume expiration date to today's date. You can identify these volumes using the DFSMSrmm search volume dialog and sort the list of volumes by the expiration date data column. However, it may be more practical to use the DFSMSrmm report extract file and process it with a tool like DFSORT to select just those volume records with an expiration date later than today's date.

   Check that the volumes concerned are covered by a VRS or that it is alright for them to be expired. If any should be retained by VRS, you should define a VRS to provide the retention that they require. If today is 1 June 2001, for each volume you can change the expiration date using the subcommand shown in Example 9-26.

Example 9-26 RMM CHANGEVOLUME to change the expiration date

```
RMM CHANGEVOLUME volser EXPDT(2001/152)
```

5. Run EDGHSKP with PARM=VRSEL and the REPORT DD statement included. Use the report output to ensure that the VRSs are now being used to provide the retention you need.
6. When the retention report is acceptable, run EDGHSKP with PARM=EXPROC. Any volume not retained by VRS will be set to pending release status. Use the DFSMSrmm SEARCHVOLUME subcommand to list all volumes in pending release status. If any volumes should not have been released, you can reclaim them by using the CHANGEVOLUME subcommand to assign a new retention period. Ensure that they are retained by a VRS before rerunning VRSEL and EXPROC processing.
7. As new retention and movement requirements are identified, use VRS exclusively to specify the retention needs.

Delete non-existent volumes

When you converted to DFSMSrmm you may have converted information about volumes that you no longer have. Some vendor tape management systems use databases that must be formatted for each range of volumes to be managed. You may have stopped using some of these ranges but never reformatted the database to remove the volumes.

Running under DFSMSrmm you have the flexibility to add and delete volumes or other resources at any time. Use the knowledge you have about your existing tape volume ranges to check that you have defined to DFSMSrmm only the volumes and rack numbers that you want to manage.

Archived

You can use the DFSMSrmm commands to build lists of commands to delete resources you do not want. When you delete volumes, consider deleting the rack numbers that are left in empty status.

For example, you know that volumes 100000 through 199999 no longer exist but that they were converted to DFSMSrmm. Use the commands in Example 9-27 to delete the volumes and rack numbers.

Example 9-27 Deleting non-existent volumes

```
RMM SEARCHVOLUME VOLUME(1*) OWNER(*) LIMIT(*) -
    NOLIST CLIST('RMM DV ',' FORCE')
EXEC EXEC.RMM LIST
RMM DELETERACK 100000 COUNT(99999)
RMM DELETERACK 199999 COUNT(1)
```

Refer to *DFSMSrmm Guide and Reference*, SC26-7404, for information about the commands used. You can repeat these commands for each range of volumes and rack numbers you want to delete.

You can avoid the need to do cleanup by deleting the volume information during conversion. Use the EDGCNXIT user exit from EDGCNVT to check for ranges of volumes you want deleted and tell EDGCNVT to skip the volumes and data set records within those ranges.

9.11.3 Implementing additional functions

As a result of your conversion, you will be using DFSMSrmm with almost the same functions as TLMS. You may wish to gain additional benefits from DFSMSrmm. These are some of the functions that you may like to implement:

- **Movement control**
  DFSMSrmm tracks the current location and destination of each volume and whether the volume is in transit to the destination. This function enables you to better track where volumes are and to use different movement cycles for each of your locations. For example, you might send volumes offsite on a daily basis to a remote location, but only weekly to a different location. DFSMSrmm provides you with the capability to control and track these movements independently.
- **Foreign and product tape management**
  DFSMSrmm can manage all the volumes in your installation, including those that do not follow your naming convention. The only volumes that cannot be defined to DFSMSrmm are those that are duplicates of volumes already defined to DFSMSrmm.

  During the conversion it is likely that all volumes were defined with rack

number identical to the VOLSERs. Now you can extend the span of control you have for tape volumes by adding to DFSMSrmm the volumes you have not previously managed. Typically, these volumes are those that are sent to you either from another site, or by a software supplier or customer.

Many of the foreign tapes will contain software products sent by the software company.

- **Notification**
  Sometimes it is useful to have the tape management system tell you of an event that has occurred or will occur. For example, you might want to know when a particular volume has been checked into the library, or that one of your volumes has expired.

  DFSMSrmm provides a notification facility that can send electronic messages to an identified user when such events occur. DFSMSrmm uses the TSO TRANSMIT command to send these messages, so they can be sent to any system that supports the receipt of messages or mail using that format.

- **Tape replacement management**
  During conversion, any permanent and temporary I/O error counts were converted to the equivalent field in DFSMSrmm. When DFSMSrmm is running it tracks permanent and temporary I/O errors that occur on tape volumes, and records the counts in the volume information. When one or more permanent errors are recorded, DFSMSrmm identifies the volume for replacement. DFSMSrmm sets the volume REPLACE release action. When the volume is no longer retained and has reached its expiration date, the volume release action must be activated.

  DFSMSrmm does not let the volume return to scratch status for reuse until the replace action has been confirmed to DFSMSrmm, and the new volume labeled.

- **Automatic CDS-driven tape erasing and labeling**
  The DFSMSrmm tape erase and initialization utility, EDGINERS, can use action information from the DFSMSrmm CDS, and it requires little or no operator intervention. You can benefit from scheduling the utility to run on a regular basis.

- **Merging and splitting your CDS**
  You should not normally have to merge or split the contents of the DFSMSrmm CDS. However, either during conversion, or when you are changing the number of MVS systems you run, you may want to merge or split the contents of the DFSMSrmm CDS.

For detailed information about how to implement these and other DFSMSrmm functions, refer to Chapter 11, “Tailoring your DFSMSrmm environment” on

page 433, and Chapter 13, “Splitting and merging your DFSMSrmm CDS” on page 561.

Archived

10

Policy management

in this chapter we describe how DFSMSrmm provides policy management for movement and retention at the data set level. Every tape data set can have a policy, and each policy can specify movement as well as retention. The retention and movement policies you define to DFSMSrmm are known as *Vital Record Specifications* (VRS). You use them to indicate how long and where you want to keep data sets or volumes. You also use them to define how volumes are to be moved among the libraries that DFSMSrmm supports, and the storage locations defined for vital records and disaster recovery purposes.

Objectives of this chapter:

- Understand the PARMLIB options for VRS support
- Learn the types of VRSs
- Learn how to create and process the VRSs

At the end of this chapter you will understand the different VRS types and how to create them. Sample VRS definitions are provided for a better understanding.

© Copyright IBM Corp. 2004. All rights reserved.

10.1 VRS related PARMLIB options

For DFSMSrmm vital record specification processing, we have several PARMLIB options that can be used for tailoring your environment:

CATRETPD Specifies the number of hours from creation that a data set should be retained if it has not been cataloged and matches a vital record specification with the WHILECATALOG operand.

MOVEBY Specify the MOVEBY operand to move volumes that are retained by DFSMSrmm vital record specifications as a set of volumes or as individual volumes.

RETAINBY Use the RETAINBY option to specify whether DFSMSrmm retains multi-volume sets as a set or as individual volumes.

REUSEBIN Specifies to control how DFSMSrmm does reuses bins when a volume is moving from a bin.

VRSCHANGE Use VRSCHANGE to specify the action that DFSMSrmm should take during inventory management if you have made any changes to vital record specifications.

VRSJOBNAME Use VRSJOBNAME to select how DFSMSrmm uses a job name in a vital record specification to match a data set to a movement and retention policy.

VRSMIN Use VRSMIN to specify a minimum number of vital record specifications required by inventory management vital record processing, and the action to be taken by DFSMSrmm when the minimum number of vital record specifications are not defined.

VRSEL Use VRSEL to control how DFSMSrmm inventory management vital record processing uses retention and movement information that are defined in vital record specifications.

For detailed information, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

10.1.1 CATRETPD

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.

You have to specify a higher catalog retention value than the default if you have a job creating data sets on tape with a long running time, and the data sets expand

a volume, or the job creates more than one data set. If the inventory management process is starting while the job is running, you cannot check if a data set is catalog or not, because the data sets are not cataloged until the job is finished. If you have not defined a catalog retention greater than the run time of your jobs, you can loose some data sets, because DFSMSrmm will release them before the job is finished.

In Figure 10-1 we describe the following situation: The default retention in the PARMLIB member is set to zero RETPD(0). A data set name vital record definition that matches all data set exists, but while cataloged only. You have started a long running job at 7:00 PM, creating one or more data sets expanding a single volume. You have not specified a retention parameter in your JCL. Early in the morning at 8:00 AM, 13 hours after your job has started, the daily EDGHSKP with the inventory management function is starting, and checks all data sets for matching vital record specifications.

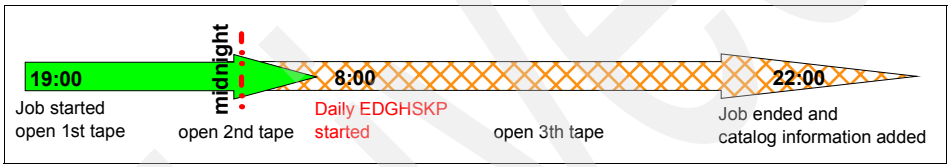


Figure 10-1 How catalog retention works

If you have not specified a CATRETPD parameter in your PARMLIB, the default value of 12 hours is used and the first volume is released, because the data set on the volume has no retention, and is not cataloged so the VRS definition does not match.

If you have specified a CATRETPD parameter in your parmlib with a value of 24 hours, the data set on the first volume is protected by this definition, and no more checks are done. The volume will not be released.

10.1.2 MOVEBY

You use this parameter to move volumes that are retained by DFSMSrmm vital record specifications as a set of volumes or as individual volumes:

- **VOLUME**
  Specify VOLUME when you want to move each volume individually.
- **SET**
  Specify SET if you want; DFSMSrmm moves a multi-volume set as an aggregate. When you retain by SET, if any volume in a set is to be moved by a vital record specification, all volumes in the set are moved to the same location.

Figure 10-2 shows how DFSMSrmm works with MOVEBY(SET) specification if you have a set of tree volumes with tree data sets on it, and two of the data sets should be moved to different locations.

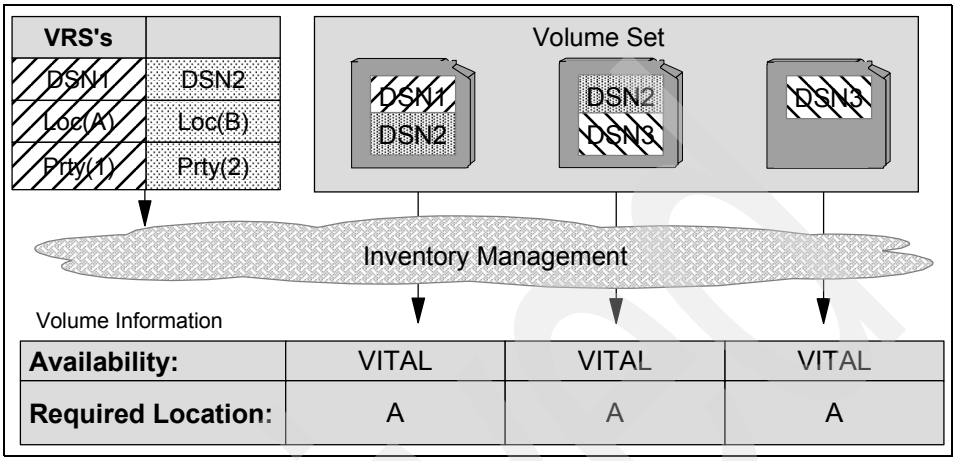


Figure 10-2 RETAINBY(SET) MOVEBY(SET)

The second volume is moved to location A because the VRS definition for data set A has a higher priority than the VRS definition for data set B. The third volume in the set is moved along with the other two volumes because it is part of the set.

10.1.3 RETAINBY

This operand allows you to work with the volume set as an aggregate:

- **VOLUME**
  Specify VOLUME when you want to retain each volume individually.
- **SET**
  Specify SET if you want DFSMSrmm to retain a multi-volume set as an aggregate. When you retain by SET, if any volume in a set is retained by a vital record specification, all volumes in the set are retained as vital records.

  When a volume in a set is only retained because it is a member of a set, DFSMSrmm sets an indicator as you can see in Figure 10-3.

```
Set retained . . . : YES
```

Figure 10-3 Retain by set indicator

Figure 10-4 show how DFSMSrmm works with the RETAINBY(SET) specification if you have a set of tree volumes with tree data sets on it.

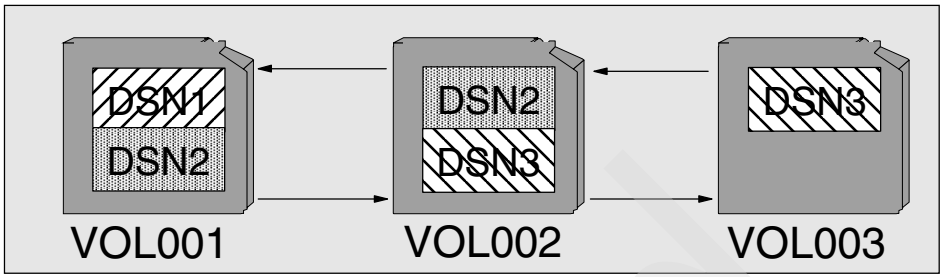
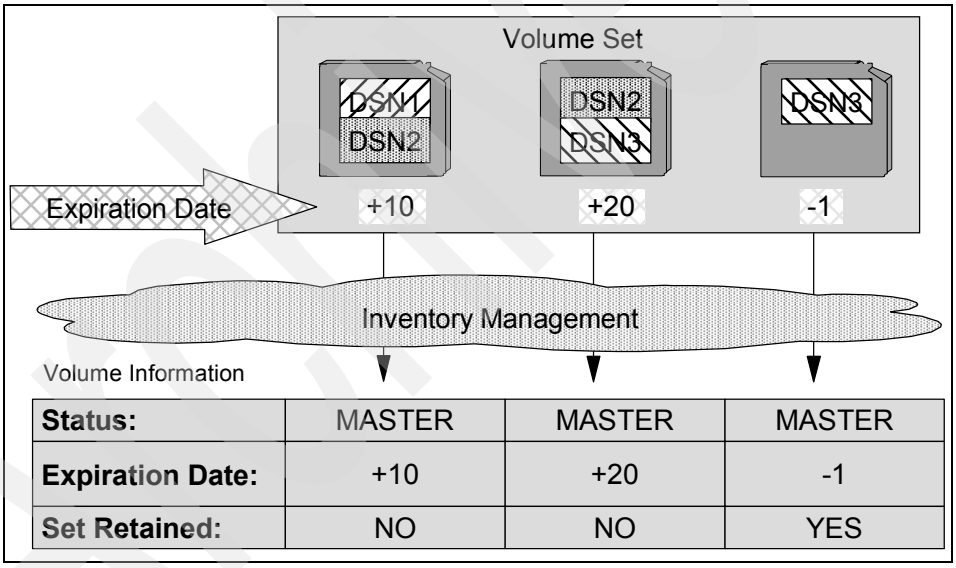


Figure 10-4 Volume set

In Figure 10-5 you can see the following: VOL001 with EXPDT +10 days, VOL002 with EXPDT +20 days and VOL003 with EXPDT -1day. After the housekeeping, all the volumes will be retained and VOL003 will set the Set Retained field to YES, because it is retained only by being a member of a set.



Status:	MASTER	MASTER	MASTER
Expiration Date:	+10	+20	-1
Set Retained:	NO	NO	YES

Figure 10-5 RETAINBY(SET) - EXPDT

If the volumes are retained by VRS, the result is shown in Figure 10-6.

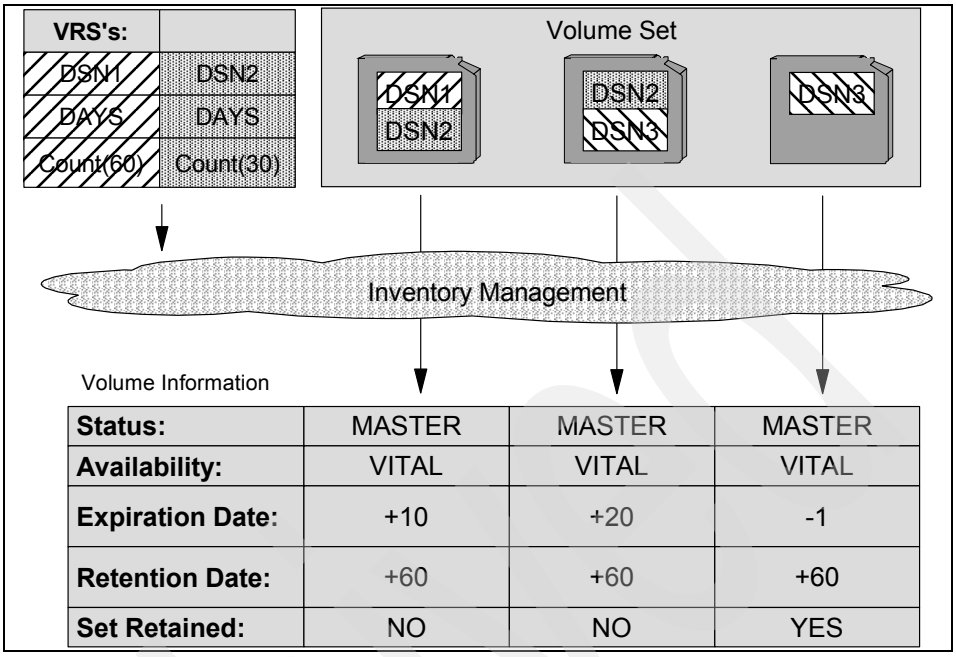


Figure 10-6 RETAINBY(SET) - VRS

10.1.4 REUSEBIN

Use this operand to control how DFSMSrmm reuses bins when a volume is moving from a bin.

- **CONFIRMMOVE**
  When a volume moves out of a bin, DFSMSrmm does not reuse this bin until the volume move has been confirmed. In this case, you have to define more bins as shown in Figure 10-7.

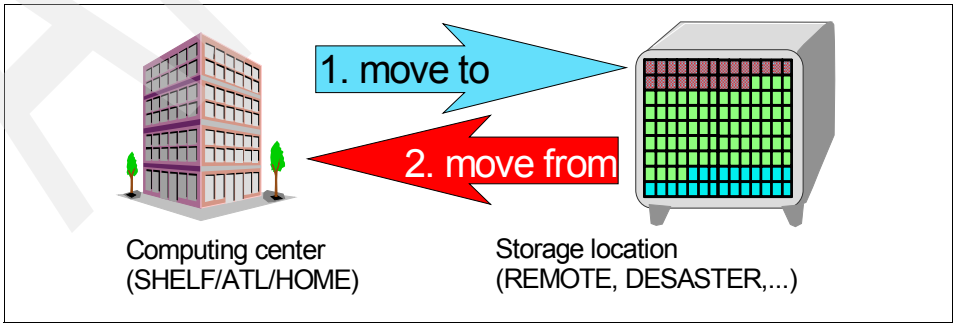


Figure 10-7 RESUSE bin with CONFIRMMOVE operand

- **STARTMOVE**
  When a volume moves out of a bin, DFSMSrmm does not reuse this bin until the volume move has been confirmed.

  A bin can be reused as soon as a volume starts moving out of a bin. Extended bin support must be enabled before you can use this operand. This parameter works only if you have enabled the extended bin support. For more information, refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405. The bins will be reused in the same processing run as shown in Figure 10-8.

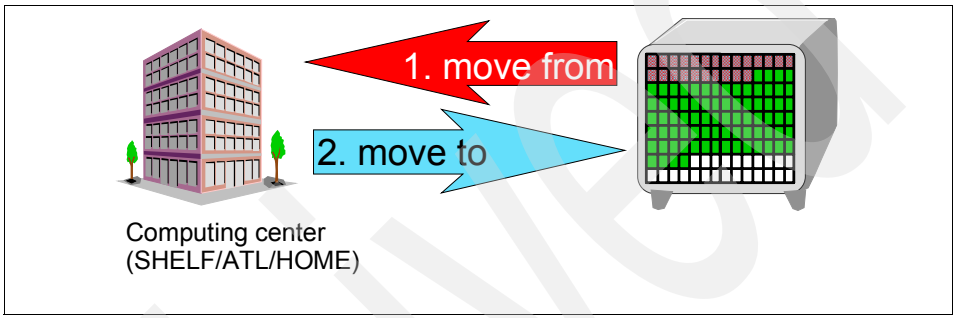


Figure 10-8 RESUSE bin with STARTMOVE operand

10.1.5 VRSCHANGE

Specify the action that DFSMSrmm should take during inventory management if you have made any changes to vital record specifications since the last run:

- **INFO**
  You get a message in the job log and no additional processing or actions are required when vital record specification changes occur.
- **VERIFY**
  Any changes made to vital record specifications must be verified by running EDGHSKP vital record processing using the VERIFY parameter. DFSMSrmm must issue a return code zero before EDGHSKP vital record processing can be performed to update the control data set.

Note: Check the ACTIFITY log of the vital record processing trial run, and check all changes made in this run before you start the life update.

Archive

10.1.6 VRSJOBNAME

Use this option to tell DFSMSrmm which of the two possible masks DSNAME or JOBNAME should be checked first:

- **1**
  If you specify one, the jobname or jobname mask will be checked first, and the data set name or data set name mask second.
- **2**
  If you specify two, the data set name or data set name mask will be checked first, and the jobname or jobname mask second.

As part of vital record processing, DFSMSrmm checks that data set name masks and job name masks used in vital record specifications are specified according to DFSMSrmm naming conventions.

DFSMSrmm compares the data set, job name, and volume information recorded in the control data set with information in the VRSs to determine which data sets and volumes to retain and the processing required. This includes any volumes with special JCL specified expiration dates used by your installation.

DFSMSrmm builds a matching string with the data set and job name. The data set and job name position depends on the value of the VRSJOBNAME parameter. DFSMSrmm selects the most specific VRS.

For example, we define the following VRSs as shown in Figure 10-9.

```
(a) RMM ADDVRS DSN('ZE*.TEST.T¬¬¬¬R¬¬') WHILECATALOG
(b) RMM ADDVRS DSN('*ERB.TEST.T¬¬¬¬R¬¬') DAYS COUNT(150)
(c) RMM ADDVRS DSN('*ERB.TEST.T¬¬¬¬R¬¬') CYCLES COUNT(10)  -
      JOBNAME('ZERB%%%')
```

Figure 10-9 VRS examples using wild card characters

We also run job ZERB0A, which creates the data set ZERB.TEST.T0023R45.

Depending on the VRSJOBNAME parameter, we get different results.

Using the **VRSJOBNAME(1)** option, we have the following syntax string as shown in Figure 10-10.

```
   ZERB0A!!ZERB.TEST.T00023R45
```

Figure 10-10 Syntax string using VRSJOBNAME(2)

In this case VRS (c) will match the string, because it matches first by job name and second by data set name.

Using **VRSJOBNAME(2)**, we have the following syntax string as shown in Figure 10-11.

```
ZERB.TEST.T00023R45!!ZERB0A
```

Figure 10-11 Syntax string using VRSJOBNAME(2)

In this case VRS (a) will match the string, because it matches by data set name having an alphanumeric character (Z) on the first position.

10.1.7 VRSMIN

Specify a minimum number of vital record specifications required by inventory management vital record processing, and the action to be taken by DFSMSrmm when the minimum number of vital record specifications are not defined. DFSMSrmm counts the vital record specifications used by vital record processing. DFSMSrmm does not count vital record specifications that are deleted during vital record processing.

- **FAIL**
  Issues message EDG2229I to the MESSAGE file and stops inventory management processing. Processing ends with return code 8.
- **INFO**
  Issues message EDG2229I to the MESSAGE file and processing continues.
- **WARN**
  Issues message EDG2229I to the MESSAGE file and sets a minimum return code of 4. Processing continues.

> **Tip:** After conversion, set the minimum number of vital record specifications you get, and set the processing action to FAIL.

10.1.8 VRSEL

Control how DFSMSrmm inventory management vital record processing uses retention and movement information that is defined in vital record specifications.

- **OLD**
  Specify OLD if you want DFSMSrmm to:
  - Ignore retention information that is defined in NAME vital record specifications
  - Ignore release options that are defined in vital record specifications

– Treat the ANDVRS in the RMM ADDVRS command as if it is a NEXTVRS

- **NEW**
  Specify NEW if you want DFSMSrmm to process:
  – Retention information in NAME vital record specifications
  – Release options defined in vital record specifications
  – Vital record specifications chained using the ANDVRS operand

The following example shows the differences between VRSEL(OLD) and VRSEL(NEW):

- We first define the following data set and management value VRSs as shown in Figure 10-12.

```
RMM ADDVRS DSNAME('YCJRES.TEST') DAYS COUNT(100) LOCATION(REMOTE)
RMM ADDVRS DSNAME('CYCL004') CYCLES COUNT(4) LOCATION(DISTANT) -
    STORENUMBER(4)
```

Figure 10-12 Define two data set name VRS definitions

- Figure 10-13 shows you some data sets created that have a management value of CYCL004 assigned.

```
(1) Data set name YCJRES.TEST creation date 120 days ago
(2) Data set name YCJRES.TEST creation date 119 days ago
(3) Data set name YCJRES.TEST creation date 118 days ago
(4) Data set name YCJRES.TEST creation date  90 days ago
(5) Data set name YCJRES.TEST creation date  89 days ago
(6) Data set name YCJRES.TEST creation date  88 days ago
```

Figure 10-13 Data set created using a management value of CYCL004

- When VRSEL(NEW) is in use, both VRSs are processed. Data sets numbers (4), (5) and (6) are retained in location REMOTE by the data set VRS, and number (3) in DISTANT location by the name VRS. DFSMSrmm retains the data set (3) by the secondary VRS (MV VRS) after the primary VRS (data set VRS) does not retain the data set any more.
- When VRSEL(OLD) is in use, only numbers (4), (5) and (6) are retained in location REMOTE. The secondary VRS is not processed.

With VRSEL(NEW), both VRSs primary and secondary are processed in parallel on each housekeeping run. DFSMSrmm evaluates both VRSs and then determines which policy to apply.

When VRSEL(NEW) is in use, the volume expiration date is not used during data set name VRS UNTILEXPIRED processing when a data set matches both a

primary VRS and a secondary VRS. For example, if a retention type of UNTILEXPIRED is defined in the primary data set name VRS, and in the secondary management value (name vital record specification) VRS there is a retention type of DAYS with COUNT(30) defined, then the volume will be retained by the secondary management value VRS. This definition retains the data set for 30 more days before being expired.

If UNTILEXPIRED is used in a secondary VRS, the volume expiration date is used to determine if the retention condition is true or not.

There is no restriction on the definition of either the primary or secondary VRS; both can be vital record specification chains containing any of the possible ADDVRS operands.

10.2 VRS types

DFSMSrmm uses policies to control the retention of data sets and volumes, and to determine a volume's required location. The retention and movement policies specified in the VRS are processed during DFSMSrmm inventory management (housekeeping).

You can define expiration and retention defaults for your installation, using the DFSMSrmm PARMLIB member EDGRMMxx.

There are three types of VRS:

- Data set VRS
- Volume VRS
- Name VRS

To define VRS, you can use DFSMSrmm ISPF panels or TSO subcommands. In the examples, we have used TSO subcommands. To define VRS using panels, you need to go to the DFSMSrmm Add Vital Record Specification panel, shown in Figure 10-14.

Archived

```
 Panel  Help
------------------------------------------------------------------------------
                  DFSMSrmm Add Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . .
       Job name mask . .

  Volume serial . .         ( May be generic )
  VRS name  . . . .
```

Figure 10-14 DFSMSrmm Add Vital Record Specification panel

10.2.1 Data set VRS

A data set vital record specification defines the 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. A data set *cycle* is a new version of the data set, so the number of data set cycles is equivalent to the number of data set versions.

You can combine or merge retention and movement policies, so you need to define more than one vital record specification for this data set. In this case, we call the *primary* vital record specification to the one that matches on:

- Data set name
- Data set name mask
- Management class name, or
- Vital record specification management value

The *secondary* vital record specification matches on:

- Management class, or
- Vital record specification management value

A vital record specification *management value* is a single qualifier name you define in a vital record specification to tell DFSMSrmm how to manage and retain your tape data sets. The vital record specification management value can be up to eight alphanumeric characters, and must begin with an alphabetic character.

Figure 10-15 shows you how DFSMSrmm performs vital record processing if you have used PARMLIB option VRSEL(NEW).

Archived

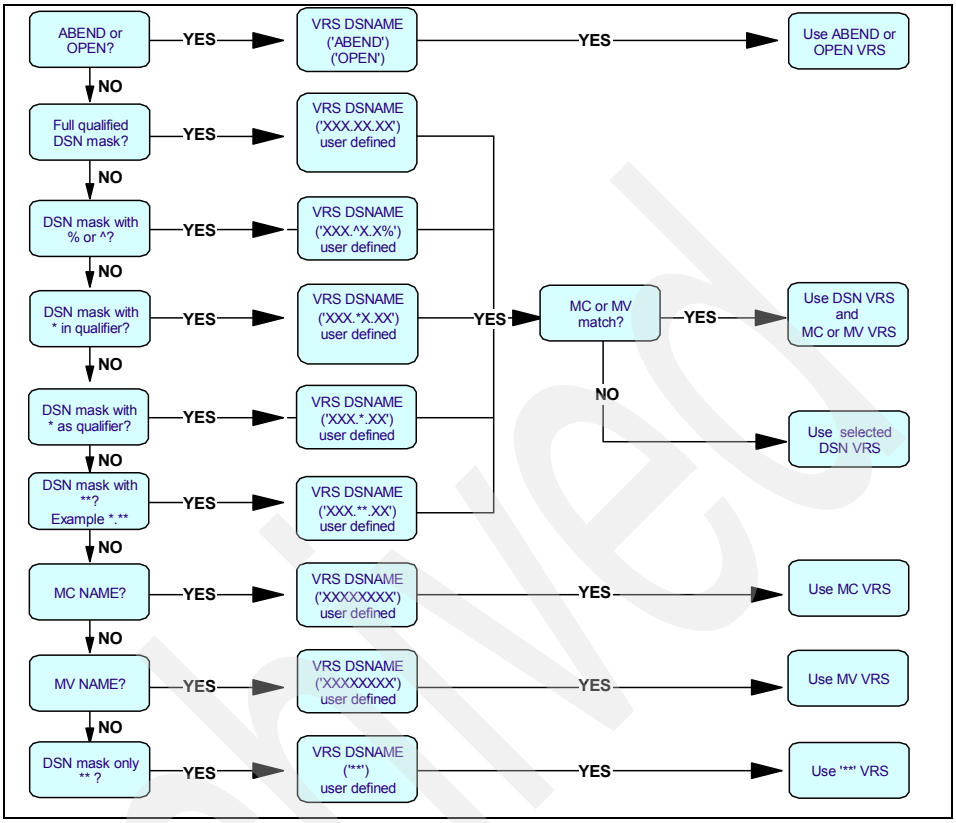


Figure 10-15 Selecting matching VRS for data sets (new)

You can choose the type of vital record processing you want DFSMSrmm to perform by specifying in the EDGRMMxx PARMLIB member, the OPTION VRSEL(OLD), or VRSEL(NEW) operand. The default is VRSEL(OLD).

Figure 10-16 shows the matching rules when using VRSEL(OLD).

Archived

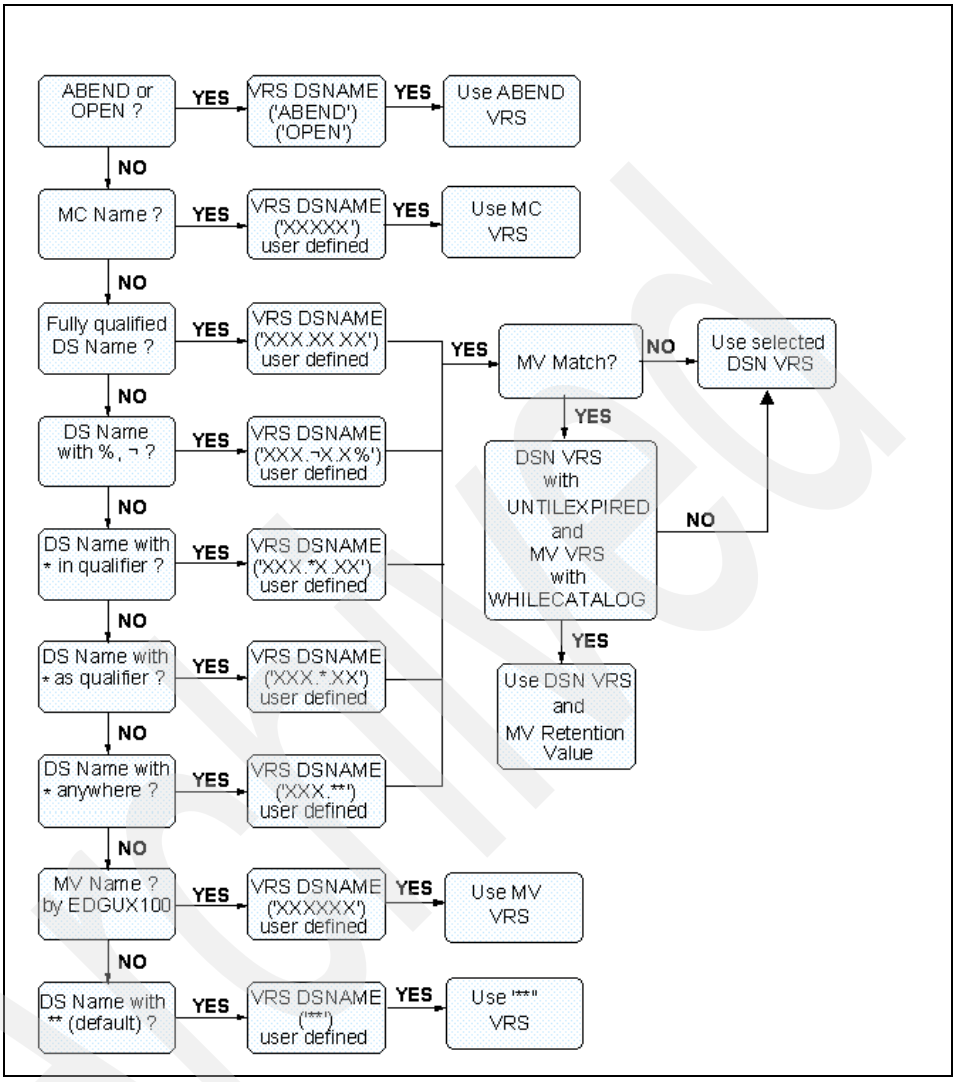


Figure 10-16 VRSEL(OLD) best match fit sequence

We recommend the use of VRSEL(NEW), because this provides new and improved functions like:

- Mixed retention types: NEXTVRS to different type
- Combined retention: ANDVRS
- Release options: EXPIRYDATEIGNORE and SCRATCHIMMEDIATE
- Enable use of primary and secondary VRSs
- Management class and VRS management value are equivalents, and can be secondary VRS policy

Archived

If you are using VRSEL(OLD), and you want to perform some new functions, you can migrate to VRSEL(NEW) as described in *DFSMSrmm Guide and Reference, SC26-7404*.

Matching data sets

Data set vital record specifications are defined using specific data set names, data set name masks, management class names, vital record specification management values, and the status of the data set. Here is the sequence for matching a tape data set name with the defined vital record specifications:

- **Fully qualified data set names/job name**
  The VRS defined in Example 10-1 retain all copies of the data set DATA.SET.BACKUP for five days. NOGDG is the default.

Example 10-1 Fully qualified VRS definition

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

- **Data set name/job name masks.**
  You can use standard wild cards: %, *, ** and ¬.

  Note: If a ¬ (not sign) is used in a data set name, the name is treated as a pseudo GDG entry.

  ** indicates to select all data sets. You can use this mask to define a vital record specification to define a 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.

- **Generation data groups (GDGs)**
  You can use data set name or data set name masks with the GDG operand. In this case, the VRS only match GDG data sets.

  The VRS definition in Example 10-2 will retain the last three versions of the GDG named DATA.SET.GDG.

Example 10-2 Define a VRS with GDG attribute

```
RMM ADDVRS DSNAME('DATA.SET.GDG') GDG CYCLE COUNT(3)
```

But, it will not retain the non-GDG data set DATA.SET.GDG. You can also define a VRS definition using the not sign (¬) wild card character to retain only GDGs as show in Example 10-3.

Example 10-3 Define a GDG using wild card characters

```
RMM ADDVRS DSNAME('DATA.SET.GDG.G¬¬¬¬V¬¬') CYCLE COUNT(3)
```

The Example 10-4 VRS definition retains all GDGs starting with DATA.TEST.

Example 10-4 Define a VRS with GDG attribute and without a limit

```
RMM ADDVRS DSNAME('DATA.TEST.**') GDG CYCLES COUNT(99999)
```

The Example 10-5 VRS definition retains all GDGs and non-GDGs starting with DATA.TEST.

Example 10-5 Define a VRS without any limit for a group of data sets

```
RMM ADDVRS DSNAME(‘DATA.TEST.**’) NOGDG CYCLES COUNT(99999)
```

- **Management class (MC)**
  You can specify a management class name in a vital record specification. The MC is assigned by the SMS ACS routines; you can find more detailed information in 10.6, “DFSMS ACS support” on page 419.

  The Example 10-6 defines a VRS using a management class name to retain a data set with MC ‘DAYS003’, this MC is assigned to the data set by your installation through an ACS routine. The operand DAYS COUNT(3) indicates the data set is to be retained for three days. Define a management class.

Example 10-6 Define a management class VRS

```
RMM ADDVRS DSNAME('DAYS003') DAYS COUNT(3)
```

Note: There is no difference in the VRS definitions between a management class and VRS management value. Only the tools (SMS or EDGUX100) setting the value are different.

- **VRS management value (MV).**
  For data sets that do not belong to a MC, but need to be managed by special expiration dates, you can define a vital record specification management value. To assign MV to data sets, use the DFSMSrmm supplied EDGUX100 installation exit.

  The Example 10-7 defines a VRS to retain a data set with special date 99000. The vital record specification MV ‘CATALOG’ is assigned by your installation in an installation exit such as the EDGUX100 user exit. The WHILECATALOG operand indicates the data set is to be retained while it is cataloged.

Example 10-7 Define a management value VRS

```
RMM ADDVRS DSNAME('CATALOG') WHILECATALOG
```

- **Status ABEND**
  To control the retention of data sets closed as a result of an abnormal end in a task, you can define a VRS using a reserved data set name mask ABEND. For example, to retain the data set one day after the last reference, use the command shown in Example 10-8.

  Note: You can define several ABEND VRSs to have different retention for data sets created using different JOBNAMEs:

  ```
  RMM ADDVRS DSNAME('ABEND') JOBNAME(*) DAYS COUNT(1)
  RMM ADDVRS DSNAME('ABEND') JOBNAME(TSM*) CYCLES COUNT(99999)
  RMM ADDVRS DSNAME('ABEND') JOBNAME(HSM*) CYCLES COUNT(99999)
  ```

  This will retain all data sets created by using the job name such as HSMA and HSMB, or TSM1 permanently, and you have to release them manually.

Example 10-8 Define an ABEND VRS

```
RMM ADDVRS DSNAME('ABEND') LASTREFERENCEDAYS COUNT(1)
```

- **Status OPEN.**
  To control the retention of data sets left open during a system failure, or that are in use during DFSMSrmm inventory management, use the reserved data set name mask OPEN. For example, to retain the data set two days after the last reference, use the command as shown in Example 10-9.

Example 10-9 Define an OPEN VRS

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

Note: You can define several OPEN VRSs to have different retention for data sets created using different JOBNAMEs:

```
RMM ADDVRS DSNAME('ABEND') JOBNAME(*)  DAYS COUNT(1)
RMM ADDVRS DSNAME('ABEND') JOBNAME(TSM*) CYCLES COUNT(99999)
RMM ADDVRS DSNAME('ABEND') JOBNAME(HSM*) CYCLES COUNT(99999)
```

This will retain all data sets created by using the job name such as HSMA and HSMB or TSM1 permanently, and you have to release them manually.

In the Figure 10-17, you can see which order DFSMSrmm matches the data set name VRSs.

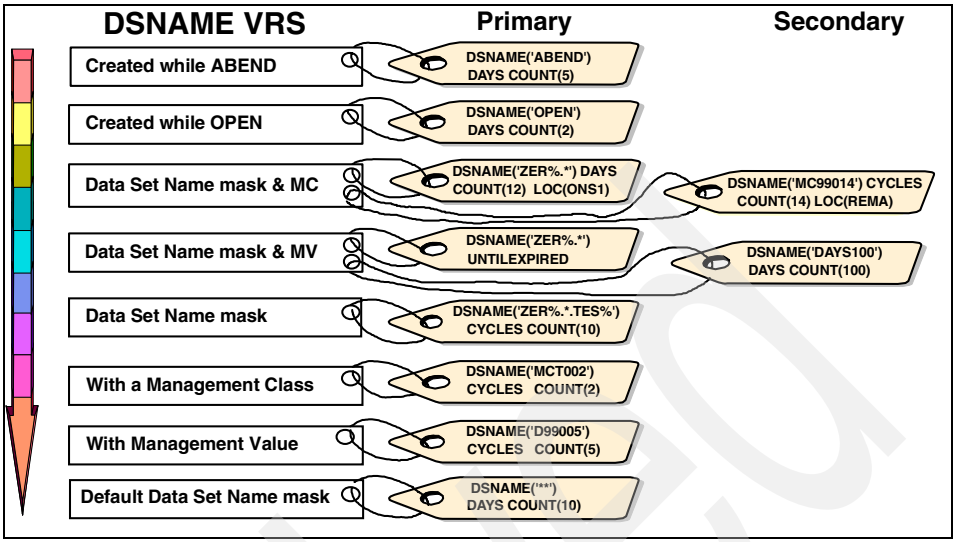


Figure 10-17 Data set name matching order

10.2.2 Volume VRS

A volume vital record specification defines the 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.

Matching volumes

Volume vital record specifications are defined using a specific or generic volume serial number. The matching rules for volumes are:

- **Specific volume serial number**
  Specify the volume serial number to retain only this volume. Example 10-10 shows you how you can retain a specific volume for five days.

Example 10-10 ADDVRS to retain a specific volume

```
RMM ADDVRS VOLUME(TST100) LOCATION(HOME) COUNT(5)
```

- **Generic volumes**
  You can specify generic volumes using “*”. Example 10-11 shows you the RMM TSO **ADDVRS** subcommand you can use to retain ten volumes matching the prefix TST.

Example 10-11 ADDVRS to retain a group of volumes

```
RMM ADDVRS VOLUME(TST*) LOCATION(REMOTE) COUNT(10)
```

10.2.3 Name VRS

Name vital record specifications can define additional movement criteria, optionally additional retention policy information. When a name VRS contains only movement information, it cannot exceed the retention criteria specified in the previous VRS, and it is called a *location* name VRS. We recommend that you do not use location name VRSs, but instead always specify a retention type and a COUNT value. You can define name vital record specifications for creating data set or volume VRS *chains*.

Specify the DFSMSrmm PARMLIB OPTION VRSEL(OLD) operand to define movement information only. Specify VRSEL(NEW) to define either retention or movement information, or both.

Example 10-12 shows the definition of a VRS chain where the second VRS is a name VRS with different retention criteria.

Example 10-12 ADDVRS chain with on subchain

```
RMM ADDVRS DSN('DATA.SET.**') COUNT(3) CYCLES NEXTVRS(SECVRS)
RMM ADDVRS NAME(SECVRS) EXTRADAYS COUNT(10)
```

For more details on chaining vital record specifications, refer to 10.4, "Chaining VRSs" on page 415.

10.3 Specifying VRS parameters

Table 10-1 shows the different parameters that you can specify in a VRS definition depending on the type of VRS that you are defining.

Table 10-1 Specifying VRS

Types of VRS & retention	DATA SET	VOLUME	NAME
Match using	- Data set name - Management class - VRS management value - GDG - ABEND - OPEN	- Volume serial number - Generic volser	

Archived

Types of VRS & retention	DATA SET	VOLUME	NAME
Retention types	- CYCLES - BYDAYSCYCLE - DAYS - LASTREFERENCE-DAYS - WHILECATALOG - UNTILEXPIRED	- Days - Volumes	- CYCLES - BYDAYSCYCLE - DAYS - LASTREFERENCE-DAYS - WHILECATALOG - UNTILEXPIRED - EXTRADAYS
Limits	- COUNT - DELETEDATE	- COUNT - DELETEDATE	- COUNT - DELETEDATE
Movement	- STORENUMBER - LOCATION - PRIORITY - DELAY	- STORENUMBER - LOCATION - PRIORITY - DELAY	- STORENUMBER - LOCATION
Release options	- SCRATCH-IMMEDIATE - EXPIRYDATE-IGNORE		
Chain using	- NEXTVRS - ANDVRS	- NEXTVRS	- NEXTVRS - ANDVRS
Chained from			- DSN VRS - VOLUME VRS - NAME VRS

10.3.1 DSNAME and NAME retention types

DFSMSrmm calculates the retention date for data sets based on a combination of the COUNT operand value and the retention type operands. The retention type operands for data sets and names are:

- **CYCLES**
  For GDG data sets, indicate the number of generations that DFSMSrmm should be kept. To retain three versions of the GDG ITSO.GDG.DATA.SET, define the VRS as shown in Example 10-13.

Example 10-13 Define a VRS with CYCLES and GDG

```
RMM ADDVRS DSNAME('ITSO.GDG.DATA.SET') GDG CYCLES COUNT(3)
```

Archived

– For non-GDG data sets, each occurrence of a data set is a cycle. Following VRS (as you can see in Example 10-14) retains three occurrences of the data set ITSO.NONGDG.DATA.SET.

Example 10-14 Define a VRS with CYCLES

```
RMM ADDVRS DSNAME('ITSO.NONGDG.DATA.SET') CYCLES COUNT(3)
```

- **BYDAYSCYCLE**
  DFSMSrmm considers all the data sets created in a single day is a cycle.
- **DAYS**
  This operand indicates a period of elapsed days from the creation date.

  Example 10-15 shows you how you can retain data sets with the high level qualifier YCJRES created in the last 10 days by issuing the following command.

Example 10-15 Define a VRS with DAYS

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

- **LASTREFERENCEDAYS**
  This operand indicates a period of elapsed days since the data set was last read or written to. You can define a VRS using Example 10-16.

Example 10-16 Define a VRS with LASTREFERENCEDAYS

```
RMM ADDVRS DSNAME('YCJRES.**') LASTREFERENCEDAYS COUNT(10)
```

  This will retain all data sets beginning by YCJRES last referenced during the last 10 days.
- **WHILECATALOG**
  You can use this operand to retain data sets while they are cataloged.
- **UNTILEXPIRED**
  A data set now optionally can be managed separately for retention. A vital record specification management value or a SMS management class can identify retention criteria, and the data set name can be used to identify movement criteria. Up to two VRSs can be used to provide policy information for a data set.

  When new data is created, and no vital record specification management value or management class is assigned, because the jobs do not specify keyword dates and are not SMS managed, VRSEL processing finds only the

Archived

data set VRS to match to, which specifies UNTILEXPIRED. Example 10-17 shows you how you can define them.

Example 10-17 Define a VRS to retain all data sets until expired

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

DFSMSrmm checks the volume expiration date, and the UNTILEXPIRED option makes DFSMSrmm drop the data set from retention.

The volume expiration date is also used if the following conditions are true and the PARMLIB option VRSEL(NEW) is specified.

- When a data set matches to a single VRS
- When UNTILEXPIRED is used in the secondary VRS

The VRS enhancements introduced with APAR OW30969 enable until expired retention to be based on any type of retention. When a primary and secondary VRS match to a data set, the use of UNTILEXPIRED in the primary uses the secondary VRS chain to determine overall retention. If there is no secondary VRS, or UNTILEXPIRED is used in a secondary VRS or a VRS management value VRS, and PARMLIB option VRSEL(NEW) is used, only the volume expiration date is used. If VRSEL(OLD) is specified in PARMLIB, UNTILEXPIRED retention uses the volume expiration date or catalog status to determine for how long to retain a data set.

Any policy used for retention or movement, or both retention and movement, can be a chain of multiple VRSs, where each VRS in a chain can use a different type of retention. Both the primary and secondary VRS can be VRS chains, offering enough flexibility to create quite complex retention and movement policies.

A secondary VRS can be only a VRS management class or a VRS management value.

10.3.2 Special NAME retention type

If you have specified to use a NEXTVRS in chain, you can use the EXTRADAYS operand on it:

- **EXTRADAYS**
  On a NEXTVRS you can specify 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. This option can only be used on a NEXTVRS and not a primary VRS.

10.3.3 Volume retention types

DFSMSrmm calculates the retention date for volumes based on a combination of the COUNT operand value and the retention type operands. The retention type operands for volumes are:

- **BY DAYS**
  By number of elapsed days since creation, which indicates that a volume is retained until the specified number of days since creation has elapsed.

  To retain volume V20001 for 31 days in a storage location, issue the command shown in Example 10-18.

Example 10-18 Define a volume VRS for a single volume

```
RMM ADDVRS VOLUME(V20001) DAYS COUNT(31) LOCATION(DISTANT)
```

- **BY NUMBER OF VOLUMES**
  Indicates how many volumes should be retained.

  For example, to retain the latest 50 volumes matching the generic volume serial number, TST, in a storage location, issue the command shown in Example 10-19.

Example 10-19 Define a volume VRS for a group of volumes

```
RMM ADDVRS VOLUME(TST*) COUNT(50) LOCATION(DISTANT)
```

10.3.4 Retention limits

You can define the limits for a volume retention or for a VRS to be considered as valid. The parameters that you can use to specify the limits are:

- **COUNT**
  Specifies a retention amount, based on the retention type chosen: number of days or cycles for data sets VRS, and number of volumes for volume VRS. The COUNT default is 99999.

  You can specify COUNT with DAYS to indicate the number of days that DFSMSrmm will retain your data set, as shown in Example 10-20:

Example 10-20 Defining a data set VRS using COUNT and DAYS

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

Archived

Or, you can specify COUNT with CYCLES to indicate the number of cycles of the data set that DFSMSrmm will retain, as shown in Example 10-21.

Example 10-21 Defining a data set VRS using COUNT and CYCLES

```
RMM ADDVRS DSNAME('YCJRES.BACKUP.CYCLE') CYCLES COUNT(8)
```

Note: A value of 99999 indicates that DFSMSrmm retains all cycles of a data set, or all volumes specified.

- **DELETEDATE**
  Specifies the date when DFSMSrmm deletes the VRS. The default value is 1999/365, which indicates that DFSMSrmm never deletes the vital record specification. When that date is reached, all matching data sets and volumes are eligible for matching to other, less specific VRSs, and if no longer VRS is retained, the volumes are eligible for release processing.

  See Example 10-22. How you can define a VRS retains the data set until the VRS deletion date (2005/170) is reached, or the data set is uncataloged.

Example 10-22 ADD a VRS definition with DELETEDATE operand

```
RMM ADDVRS DSNAME('YCJRES1.DELETE.DATE')     -
           DELETEDATE(2005/170) WHILECATALOG
```

Note: If you have a DELETEDATE specified, and the date is reached, all data sets matching this VRS will never be passed to another VRS.

10.3.5 Movement policies

You can define policies to provide retention information for data sets and volumes that must be moved through multiple locations before they expire. You define such policies by creating a VRS chain, and using the following parameters:

- **STORENUMBER**
  For data set VRS, this specifies how many days to retain a data set, how many data set cycles or versions to retain in the location where the volume is. For volume VRS specifies how many volumes to retain, or how many days to retain a volume in the location specified in the parameter LOCATION. The number specified must be less than or equal to COUNT.
- **LOCATION**
  Specifies a location where the volume should be retained. Use HOME if you want the volume to be returned to its home location. Use CURRENT to avoid moving volumes.

- **PRIORITY**
  When multiple data sets on the same volume are retained by VRSs, and each VRS contains a different destination, DFSMSrmm decides where to move the volume according to priority number. The default value 0 means that the priority specified in the LOCDEF parameter on PARMLIB should be used.
- **DELAY**
  Specifies the number of days that a data set or volume is retained in its current location before it is sent to a storage location.

10.3.6 Release options

Specifies special release options to be processed after a data set is dropped from VRS, and the volume is released. Release options take effect only for data sets once they have been retained by a VRS.

- **SCRATCHIMMEDIATE**
  If you specify this option, DFSMSrmm will return the volume to scratch status in a single run of inventory management (EXPROC processing) when the only pending release action for the volume is to return to scratch, and you have specified VRSEL(NEW).
- **EXPIRYDATEIGNORE**
  If you specify this option, DFSMSrmm ignores the volume expiration date when VRSEL(NEW) is used and no other vital VRS matches to the data set.

10.4 Chaining VRSs

A *VRS chain* is composed by a volume or data set VRS, and all of the name VRSs chained from it. For chaining two or more VRS, you must use one of the two following operands:

- **NEXTVRS**
  Specify NEXTVRS(name) to point to another VRS, and create a chain of VRSs. With NEXTVRS, DFSMSrmm processes each vital record specification in the chain separately.

  You can only chain volume VRSs with NEXTVRS operand.
- **ANDVRS**
  Specify ANDVRS to create a chain of VRSs where all the retention conditions

Archived

must be true for the data sets to be retained. DFSMSrmm uses the STORENUMBER and LOCATION from the first VRS in the VRS chain.

You can combine VRSs using NEXTVRS and ANDVRS operands; these combinations are called VRS chain and VRS subchain:

- VRS chain is a volume or data set VRS, and all the name VRSs chained from it.
- VRS subchain is a data set VRS, name VRS with retention information, or ANDVRS group, and all the VRSs chained from it, without including the NEXTVRS in the chain that contains retention information.

Both a VRS chain and subchain may comprise one or more VRSs. When a data set name VRS is chained only to name VRSs that have no retention information, there is no difference between a VRS chain and a VRS subchain. The following examples explain the differences between chain and subchain VRS.

Figure 10-18 shows a VRS chain with three subchains, they are a data set VRS with NEXTVRS, and two-name VRSs with retention information.

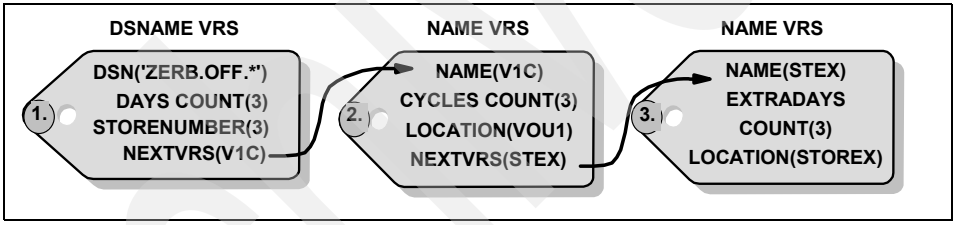


Figure 10-18 VRS chain with three subchains

Figure 10-19 shows a VRS chain with two subchains, it is a data set VRS with a ANDVRS that form one subchain, and a name VRS joined with **NEXTVRS** command with retention information.

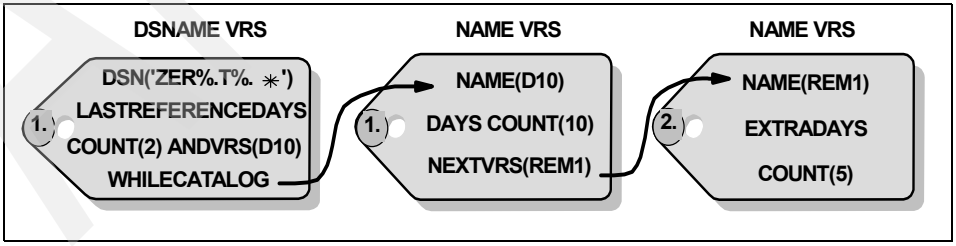


Figure 10-19 VRS chain with two subchains

Figure 10-20 shows a VRS chain with a single subchain; the VRS are joined by ANDVRS commands.

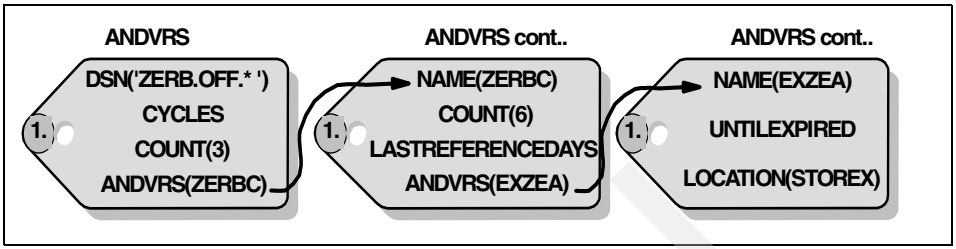


Figure 10-20 VRS chain

Figure 10-21 show a VRS chain with two subchains: the data set VRS and the first name VRS form a subchain, because the name VRS does not have retention information. The second name VRS is a different VRS subchain.

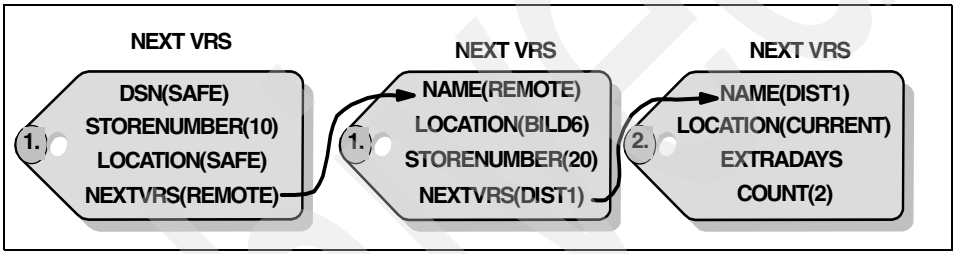


Figure 10-21 VRS chain with two subchains

10.4.1 VRS chain and subchain

The VRS is key to how DFSMSrmm manages your tape data sets and volumes; retention and movement are driven by VRSs. There are many ways to build VRS chains and subchains. Even if you are able to meet all your retention and movement needs, your approach should be to keep the VRS chain and subchain as simple as possible.

There are some terms you should be familiar with before we discuss building VRS chains and subchains:

Data Set Group All the data sets with a name that matches a VRS are treated as a single vital retention group. You can use a GDG base name when defining a VRS 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 data set group. If you are using GDG version numbering, DFSMSrmm only keeps the latest version of each generation.

VRS chain A volume or data set VRS and all of the name VRSs chained from it

VRS subchain A data set name VRS, NAME VRS with retention information, or AND VRS group and all the VRSs chained from it, up to but not including the next VRS in the chain that contains retention information. Both a VRS chain and subchain may comprise one or more VRSs. When a data set name VRS is chained only to NAME VRSs that have no retention information, there is no difference between a VRS chain and a VRS subchain.

Note: A data set VRS can only be used as the first definition in a chain. A data set VRS cannot be used in the NEXTVRS operand.

10.5 Managing DFSMShsm tapes

The interface between DFSMShsm and DFSMSrmm is automatically in place when you install DFSMS, so you do not need to take any action to activate it, or to call it.

You can use DFSMSrmm PARMLIB OPTION TVEXTPURGE operand to control the processing that DFSMShsm performs when it releases a volume. You can set this volume to pending release or set the volume expiration date to the current date. These are the different options for this operand:

- **NONE**
  DFSMSrmm takes no action for volumes to be purged.
- **RELEASE**
  This is the default and the only action if DFSMSrmm is working with the VRSEL(OLD) operand. With this the operand uses the VRS as shown in Example 10-23 to retain DFSMShsm data sets.

Example 10-23 ADD a VRS to retain DFSMShsm managed data sets

```
RMM ADDVRS DSNAME('mprefix.HMIGTAPE.DATASET') COUNT(99999) CYCLES
RMM ADDVRS DSNAME('mprefix.BACKTAPE.DATASET') COUNT(99999) CYCLES
RMM ADDVRS DSNAME('mprefix.COPY.HMIGTAPE.DATASET') COUNT(99999) CYCLES
RMM ADDVRS DSNAME('mprefix.COPY.BACKTAPE.DATASET') COUNT(99999) CYCLES
```

These VRSs protect all your volumes until DFSMShsm releases the tapes; in this case the volumes are set to pending release, and with the next expiration processing, to scratch.

- **EXPIRE.**
  Use the EXPIRE option to set the volume expiration date to the current date

for volumes released by DFSMShsm. Use this operand in combination with vital record specifications that uses the UNTILEXPIRED retention type. You can then optionally extend retention using the extra days retention type.

By default, DFSMShsm uses 1999/365 as the expiration date for permanent retention. When DFSMShsm releases a tape, DFSMSrmm sets the current date as the expiration date.

For example, to retain DFSMShsm migration data sets three more days after DFSMShsm releases it:

- Specify the PARMLIB OPTION VRSEL(NEW).
- Specify the PARMLIB OPTION TVEXTPURGE(EXPIRE).
- Add the following VRS as shown in Example 10-24.

Example 10-24 ADD a VRS to expand DFSMShsm release actions

```
RMM ADDVRS DSNAME('mprefix.HMIGTAPE.DATASET') UNTILEXPIRED -
    NEXTVRS(HSMEXT)
RMM ADDVRS NAME(HSMEXT) EXTRADAYS COUNT(3)
```

10.6 DFSMS ACS support

Before OS/390 V2R10, you could assign a management value to the data set created on a any tape volume. For doing this, DFSMSrmm calls to EDGUX100 exit at O/C/EOV time to assign a MV to the data set. During VRSEL processing, this MV assigned to the data set is used to assign a VRS. Figure 10-22 shows this process. EDGUX100 also can be used to modify the WTO mount message.

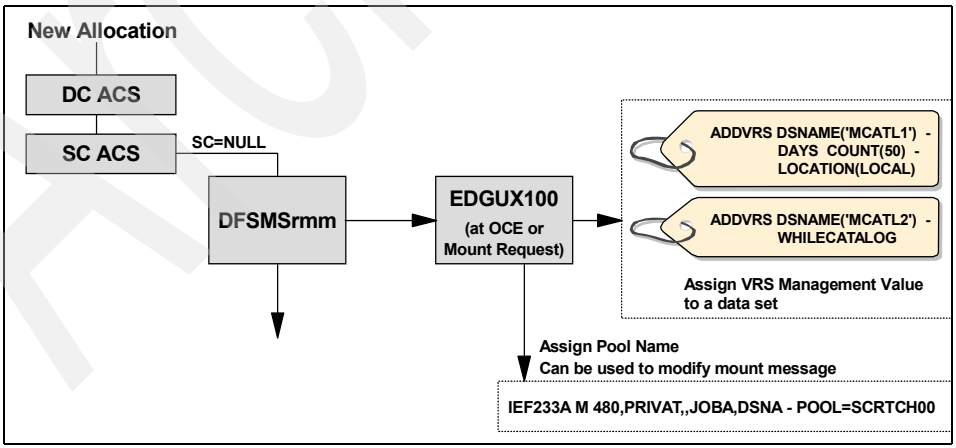


Figure 10-22 Support before OS/390 V2R10 for non-system-managed tapes

Archived

Some enhancements were introduced with DFSMSdfp prior to OS/390 R10 to add pre-ACS support using the SMS exit IGDACSXT, which gets control before the ACS routines. On OS/390 R10 DFSMSrmm added both pre-ACS support to support the DFSMSdfp processing, and DFSMSrmm ACS support to enable you to use the management class for VRS policy management.

For more details about this function and how to implement this enhancement, refer to Chapter 4., “DFSMSrmm ACS support” on page 137.

10.7 Modifying VRS

All the modifications to VRS definitions can be made by TSO subcommands or by DFSMSrmm ISPF panels. Here we are using only DFSMSrmm ISPF panels. To see a description of TSO subcommands, refer to the *DFSMSrmm Guide and Reference, SC26-7404.*

DFSMSrmm does not provide a RMM TSO **CHANGEVRS** subcommand for modifying an existing VRS. You can use a combination of DELETEVRS and ADDVRS subcommands for this. Another way is to use the ISPF dialog.

10.8 Deleting VRS

Before deleting a VRS, you must be sure that you want all the data set or volumes protected by this VRS to be released. You can verify this in the DFSMSrmm VRSs panel (Figure 10-23). Select **M** before the VRS you are planning to delete and press Enter.

```
                        DFSMSrmm VRSs (Page 1 of 3)        Row 1 to 10 of 10
Command ===>                                                Scroll ===> CSR

The following line commands are valid: C, D, L, M, O, S.
Use the RIGHT command to view other data columns.

S Volume/Data set/Name specification           Job name Type Location Priority
- ---------------------------------------------- -------- ---- -------- --------
  *                                                      DSN  HOME     0
  D99000                                              DSN  HOME     0
  M98031                                              DSN  HOME     0
  OPEN                                                   DSN  HOME     0
  SAFE                                                DSN  REMOTE   0
M YCJRES%.T*.*                                        DSN  HOME     0
  YCJRES1.CAT*.**                                     DSN  HOME     0
```

Figure 10-23 DFSMSrmm VRSs panel

DFSMSrmm shows all the data sets matched by this VRS (Figure 10-24) and you can check if you want to release all of them. The listed data sets may be retained by other more specific VRSs; you can check which VRSs retains a data set by using the `L` line command from the returned list.

```
                          DFSMSrmm Data Sets               Row 1 to 4 of 4
Command ===>                                              Scroll ===> CSR


The following line commands are valid: C, D, I, L, O, V.
                                                 Volume          File Create
S Data set name                                  Serial Owner    Seq  Date
- ----------------------------------------------- ------ -------- ---- ----------
  YCJRES1.TEST1.VRS2                             TST016 YCJRES1  1    2001/059
  YCJRES1.TEST1.VRS2                             TST023 YCJRES1  1    2001/060
```

Figure 10-24 DFSMSrmm Data Sets panel

To delete the VRS, you can use the ISPF dialog, or use the DELETEVRS command.

10.9 Trial run

DFSMSrmm provides the EDGHSKP utility to help you perform inventory management. When you specify the VERIFY parameter, then DFSMSrmm performs a trial run of vital record processing, so you can see how vital record processing changes affect the data set and volume information before the CDS is updated. Figure 10-25 contains a sample JCL for inventory management trial run processing.

```
//HSKP EXEC PGM=EDGHSKP,PARM='VRSEL,VERIFY,DATE(+5)'
//MESSAGE   DD  DISP=SHR,DSN=HSKP.MESSAGES
//REPORT    DD  DISP=SHR,DSN=HSKP.VRS.REPORT
//ACTIVITY  DD  DISP=SHR,DSN=HSKP.ACTIVITY
```

Figure 10-25 Inventory management trial run

The parameter DATE(+5) adds 5 days to current day to perform a trial run in this date.

The activity file is not intended to be a report, it just contains information about changes made to data sets during vital record processing. DFSMSrmm provides a sample job EDGJACTP in SAMPLIB that shows how to selectively format and print fields.

Archived

The output for this job are nine data sets containing the following reports:

- RUNINFO report. Data set with run control information
- VRS report. Data sets with changed V/R status
- VRSS report. Summary by status
- RETDATE report. Data sets with new retention date
- RETDS report. Summary by new date
- MATCHVRS report. Data sets with new matching VRS
- MATCHVS report. Summary by new matching VRS
- SUBCHN report. Data sets retained by new part of chain
- SUBCHNS report. Summary by new retaining subchain

For more information about these reports and the EDGJACTP job, please refer to *DFSMSrmm Reporting,* SC26-7406.

10.10 VRS examples

The following are examples of a VRS chain and subchain to explain the difference between these concepts.

10.10.1 ADDVRS operand defaults

In the examples, we use the default values for the following operands of the **ADDVRS** subcommand:

Operand	Default value
COUNT	99999
DELAY	0
DELETEDATE	1999/365
LOCATION	HOME
PRIORITY	0
STORENUMBER	99999

10.10.2 Adding a single VRS

In the first example (see Figure 10-26), we are using a single VRS to retain and move a data set. The data sets shown in Figure 10-27 are retained only by this PRIMARY VRS.

The tasks for this DSN VRS are as follows:

- Retain ten cycles of each matching data set group.
- All instances of a data set created on a single calendar day are considered to be a single cycle.

Archived

- The newest five cycles must be stored in the outside storage location SAVE, and all others in their HOME location.

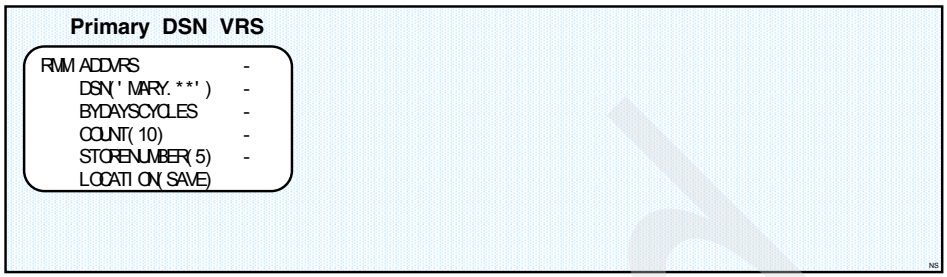


Figure 10-26 Single VRS example

Figure 10-27 shows the data sets and the location retained by the single VRS definition.

data set name	creation date	location	creation occurrence	
MARY.TSO.DATA.BACKUP	2002/123	HOME	-9	
MARY.TSO.DATA.BACKUP	2002/124	HOME	-8	
MARY.TSO.DATA.BACKUP	2002/125	HOME	-7	
MARY.TSO.DATA.BACKUP	2002/126	HOME	-6	
MARY.TSO.DATA.BACKUP	2002/127	HOME	-5	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/127	HOME	-5	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/130	SAVE	-4	
MARY.TSO.DATA.BACKUP	2002/131	SAVE	-3	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/131	SAVE	-3	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/132	SAVE	-2	
MARY.TSO.DATA.BACKUP	2002/133	SAVE	-1	
MARY.TSO.DATA.BACKUP	2002/134	SAVE	0	

Figure 10-27 Single VRS Retention and location results

10.10.3 VRS chain with one NEXTVRS operand

In the second example (Figure 10-28), we are using the NEXTVRS operand to have an additional LOCATION definition. The request for this VRS is almost the same as the one in Figure 10-26; the difference is that the newest five cycles must always be in the HOME location.

The tasks for this VRS specification are as follows:

- Retain ten cycles of each matching data set group.
- All instances of a data set created on a single calendar day are considered to be a single cycle.

- The newest cycle (0) must be stored in the HOME location. You do not need to specify the `LOCATION(HOME)` operand, because HOME is the default.
- The next five cycles (-1 to -5) must be stored in the outside storage location SAVE.
  - All other cycles (-6 to -9) should be retained in the HOME location.

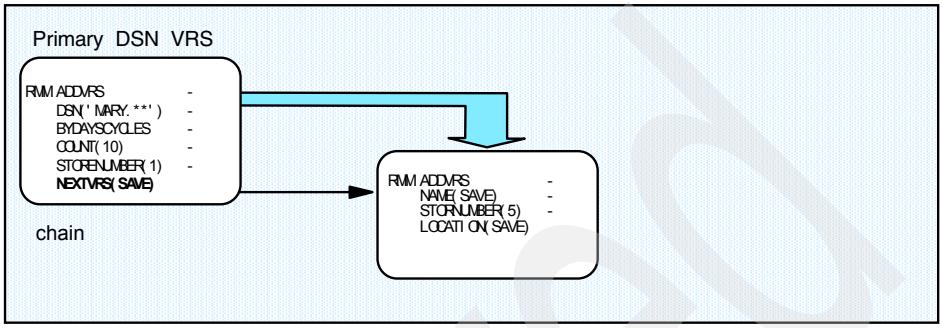


Figure 10-28 VRS chain with one NEXTVRS operand

Figure 10-29 shows the data sets and their location retained by this VRS definition.

data set name	creation date	location	creation occurrence	
MARY.TSO.DATA.BACKUP	2002/123	HOME	-9	
MARY.TSO.DATA.BACKUP	2002/124	HOME	-8	
MARY.TSO.DATA.BACKUP	2002/125	HOME	-7	
MARY.TSO.DATA.BACKUP	2002/126	HOME	-6	
MARY.TSO.DATA.BACKUP	2002/127	SAVE	-5	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/127	SAVE	-5	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/130	SAVE	-4	
MARY.TSO.DATA.BACKUP	2002/131	SAVE	-3	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/131	SAVE	-3	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/132	SAVE	-2	
MARY.TSO.DATA.BACKUP	2002/133	SAVE	-1	
MARY.TSO.DATA.BACKUP	**2002/134**	**HOME**	**0**	

Figure 10-29 Single VRS using the NEXTVRS operand results

VRS chain with two NEXTVRS definitions

The third example (see Figure 10-30) shows a VRS with two NEXTVRS definitions to have more than one outside LOCATION definition. The only difference between the first and the second example is that the data sets must be moved to a secondary storage location.

Archived

The tasks for this VRS specification are as follows:

- Retain ten cycles of each matching data set group.
- All instances of a data set created on a single calendar day are considered to be a single cycle.
- The newest cycles (0) must be stored in the HOME location. You do not need to specify the `LOCATION(HOME)` operand, because HOME is the default.
- The next three cycles (-1 to -3) must be stored in the outside storage location SAVE.
- The cycles (-4 to -6) must be stored in the outside storage location MUNICH.
- All other cycles (-7 to -9) should be retained in the HOME location.

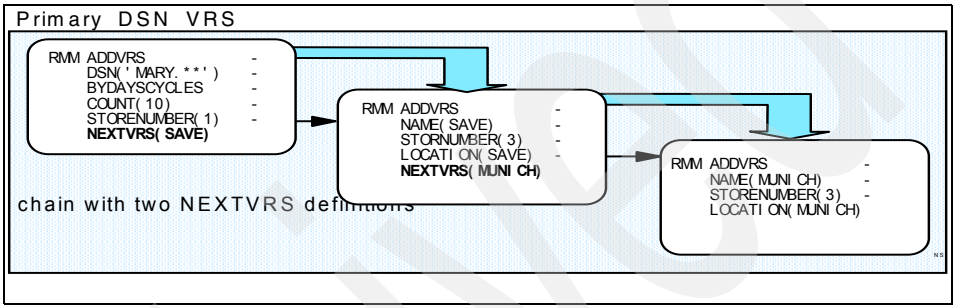


Figure 10-30 VRS chain with Two NEXTVRS operands

Figure 10-31 shows the data sets and their location retained by this VRS definition.

data set name	creation date	location	creation occurrence	
MARY.TSO.DATA.BACKUP	2002/123	HOME	-9	
MARY.TSO.DATA.BACKUP	2002/124	HOME	-8	
MARY.TSO.DATA.BACKUP	2002/125	HOME	-7	
MARY.TSO.DATA.BACKUP	**2002/126**	**MUNICH**	**-6**	
MARY.TSO.DATA.BACKUP	**2002/127**	**MUNICH**	**-5**	**BYDAYSCYCLE**
MARY.TSO.DATA.BACKUP	**2002/127**	**MUNICH**	**-5**	**BYDAYSCYCLE**
MARY.TSO.DATA.BACKUP	**2002/130**	**MUNICH**	**-4**	
MARY.TSO.DATA.BACKUP	2002/131	SAVE	-3	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/131	SAVE	-3	BYDAYSCYCLE
MARY.TSO.DATA.BACKUP	2002/132	SAVE	-2	
MARY.TSO.DATA.BACKUP	2002/133	SAVE	-1	
MARY.TSO.DATA.BACKUP	2002/134	HOME	0	

Figure 10-31 Single VRS using two NEXTVRS operand results

VRS chain with two NEXTVRS operands with retention information

Figure 10-32 shows a management class (MC) or a management value (MV) VRS using two NEXTVRS operands to have more than one retention criterion. All data sets retained by this VRS are retained in the HOME location. This example is a VRS chain with two additional subchains.

The tasks for this VRS specification are as follows:

- Retain a data set for five calendar days by specifying the operand DAYS.
- After five calendar days, retain the data set as long as it is cataloged by specifying the `WHILECATALOG` operand.
- If the data set is no longer cataloged, retain the data set for one additional day by specifying the `EXTRADAYS` operand.

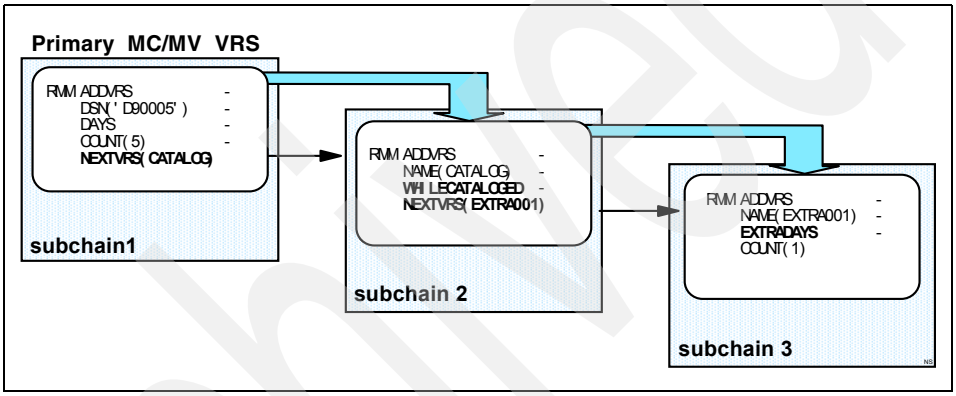


Figure 10-32 VRS chain with retention information example

In Figure 10-33, the generation data group (GDG) is defined with a limit of 10. Generation data set MARY.TSO.DATA.G0011V00 was uncataloged manually yesterday, and the generation data set MARY.TSO.DATA.G0010V00 was uncataloged manually last week. The first of the two MARY.TSO.DATA.G0013V00 generation data sets was created incorrectly and not cataloged.

Archived

data set name	creation date	location	data set cataloged	retention criteria
MARY.TSO.DATA.G0009V00	2002/123	HOME	Y	WHILECATALOGED
MARY.TSO.DATA.G0010V00	2002/124	HOME	N	not retained
MARY.TSO.DATA.G0011V00	2002/125	HOME	N	EXTRADAYS
MARY.TSO.DATA.G0012V00	2002/126	MUNICH	Y	WHILECATALOGED
MARY.TSO.DATA.G0013V00	2002/127	MUNICH	N	not retained
MARY.TSO.DATA.G0013V00	2002/127	MUNICH	Y	WHILECATALOGED
MARY.TSO.DATA.G0014V00	2002/130	MUNICH	Y	DAYS
MARY.TSO.DATA.G0015V00	2002/131	SAVE	Y	DAYS
MARY.TSO.DATA.G0015V00	2002/131	SAVE	Y	DAYS
MARY.TSO.DATA.G0016V00	2002/132	SAVE	Y	DAYS
MARY.TSO.DATA.G0017V00	2002/133	SAVE	Y	DAYS
MARY.TSO.DATA.G0018V00	2002/134	HOME	Y	DAYS

Figure 10-33 Single VRS using two retention NEXTVRS operands results

10.10.4 VRS chain with two ANDVRS operands

The fifth example (see Figure 10-34) shows a complex VRS with two ANDVRS operands in order to have more than one retention criterion for retaining a data set. If you specify ANDVRS, all definitions must be true to retain the data set. If any definition is not true, the data set is no longer retained by this VRS chain.

The tasks for this VRS are as follows:

- Retain a data set for three cycles.
- Only do so if the data set has been used in the last six days.
- Also, the data set should be moved to the outside storage location STOREX until the volume expiration date is reached.

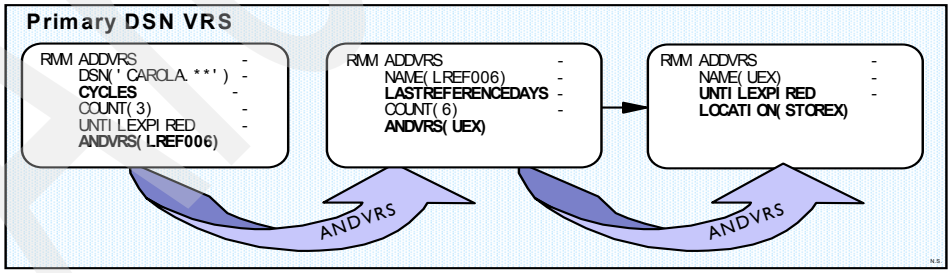


Figure 10-34 VRS chain with two ANDVRS operands example

Figure 10-35 shows the data sets and their location retained by the VRS definition in Figure 10-34. Only the generation data sets G0016V00, G0017V00, and G0018V00 are retained by the VRS because we specified CYCLES COUNT(3).

data set name	creation date	exp. date	Last used date	retention criteria			
				WC	LR	UEX	
MARY.TSO.DATA.G0009V00	2002/123	2002/123	2002/123	N	N	N	
MARY.TSO.DATA.G0010V00	2002/124	2002/124	2002/124	N	N	N	
MARY.TSO.DATA.G0011V00	2002/125	2002/125	2002/231	N	N	N	
MARY.TSO.DATA.G0012V00	2002/126	2002/126	2002/126	N	N	Y	
MARY.TSO.DATA.G0013V00	2002/127	2002/127	2002/127	N	N	Y	
MARY.TSO.DATA.G0013V00	2002/127	2002/127	2002/132	N	Y	Y	
MARY.TSO.DATA.G0014V00	2002/130	2002/130	2002/130	N	N	Y	
MARY.TSO.DATA.G0015V00	2002/131	2002/131	2002/100	N	Y	Y	
MARY.TSO.DATA.G0015V00	2002/131	2002/131	2002/131	N	N	Y	
MARY.TSO.DATA.G0016V00	2002/132	2002/132	2002/132	Y	Y	Y	<===
MARY.TSO.DATA.G0017V00	2002/133	2002/133	2002/133	Y	Y	Y	<===
MARY.TSO.DATA.G0018V00	2002/134	2002/134	2002/134	Y	Y	Y	<===

Figure 10-35 Single VRS using two retention NEXTVRS operands results

10.10.5 VRS chain with two subchains

Figure 10-36 shows a complex VRS with an ANDVRS definition, and a NEXTVRS VRS with retention information. The tasks for this VRS are as follows:

- Retain a data set for three cycles.
- Only do so if the data set has been used in the last six calendar days.
- If the data set is no longer retained by the first subchain, the data set must be retained for five additional days.

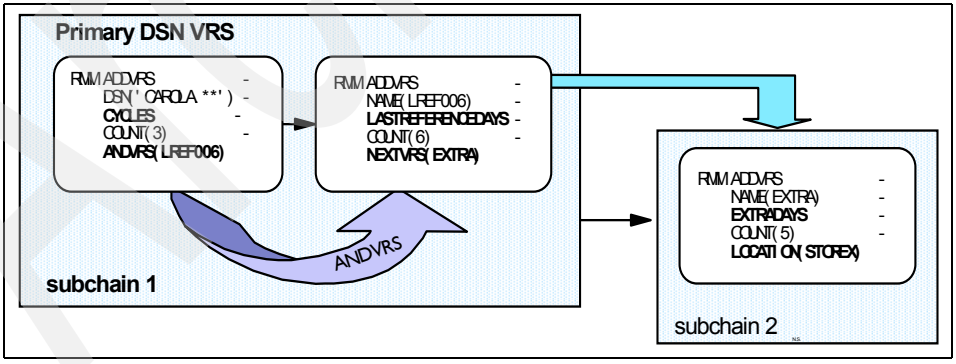


Figure 10-36 VRS chain with Two Subchains Example

Figure 10-37 shows the data sets and their location retained by the VRS definition in Figure 10-36. The generation data sets G0016V00, G0017V00, and G0018V00 are retained by the first subchain, and the generation data set

G0015V00 is retained and moved to the storage location STOREX by the secondary subchain.

```
data set name           creation  exp.      Last useed  retention
                        date      date      date        criteria
                                                        WC  LR       *
MARY.TSO.DATA.G0009V00  2002/123  2002/123  2002/123    N   N
MARY.TSO.DATA.G0010V00  2002/124  2002/124  2002/124    N   N
MARY.TSO.DATA.G0011V00  2002/125  2002/125  2002/231    N   N
MARY.TSO.DATA.G0012V00  2002/126  2002/126  2002/126    N   N
MARY.TSO.DATA.G0013V00  2002/127  2002/127  2002/127    N   N
MARY.TSO.DATA.G0013V00  2002/127  2002/127  2002/132    N   N
MARY.TSO.DATA.G0014V00  2002/130  2002/130  2002/130    N   N
MARY.TSO.DATA.G0015V00  2002/131  2002/131  2002/100    N   N
MARY.TSO.DATA.G0015V00  2002/131  2002/131  2002/131    N   Y   <=== 2
MARY.TSO.DATA.G0016V00  2002/132  2002/132  2002/132    Y   Y   <=== 1
MARY.TSO.DATA.G0017V00  2002/133  2002/133  2002/133    Y   Y   <=== 1
MARY.TSO.DATA.G0018V00  2002/134  2002/134  2002/134    Y   Y   <=== 1

* = retained by VRS subchain
```

Figure 10-37 ANDVRS and NEXTVRS with retention operand results

10.11 Complex VRS chain with three subchains

Figure 10-38 shows a complex VRS with a primary DSN VRS that has two NEXTVRS definitions followed by two subchains with retention information. The second subchain has a NEXTVRS definition, and the third subchain is using the ANDVRS operand.

The tasks for this VRS are as follows:

1. The newest 20 cycles are retained by the first subchain.
   - The newest five cycles are always in the HOME location.
   - The next five cycles are moved to the storage location SAVE.
   - The cycles -10 to -14 are retained in the storage location MUNICH.
   - The last five cycles are retained in the HOME location.
2. After the data sets are no longer retained by the first subchain, we will retain all data sets for 365 calendar days since creation:
   - The first 60 days, retain the data sets in the HOME location.
   - Then retain the data sets for 14 days in the storage location MAINZ.
   - The last 291 days, retain the data sets in the HOME location.
3. The third subchain retains the data set only under the following conditions:

– If we have up to 30 cycles.
– *And* if the data set has been read or written to in the last 400 calendar days.

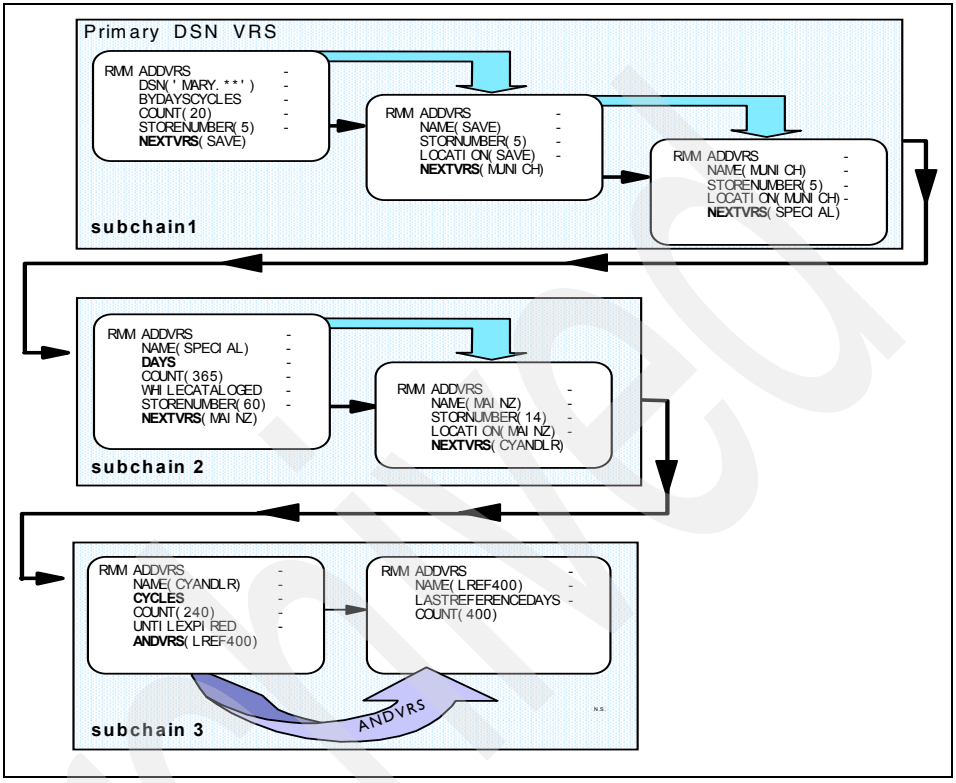


Figure 10-38 VRS chain with three subchains example

10.12 VRS hints and tips

With OS/390 V2R10 and VRSEL(NEW) parameters, system-managed and non-system-managed tape data sets can be managed using ACS routines. This allows you to simplify VRS specifications and centralize the data set specifications in the ACS routines.

You can benefit from the new VRS support introduced by VRSEL(NEW) following these recommendations:

- Move data set name masks to ACS filter list.
  If you have the VRSs as shown in Example 10-25.

Example 10-25 VRS data set name definition with the same retention criteria

```
RMM ADDVRS DSNAME('DATA.SET.**') COUNT(31) DAYS
RMM ADDVRS DSNAME('YCJRES.SET.**') COUNT(31) DAYS
```

You can assign a MC LDATE031 through ACS routines to these data set name masks, and add the following VRS as shown in Example 10-26.

Example 10-26 VRS management class definition

```
RMM ADDVRS DSNAME('LDATE031') COUNT(31) DAYS
```

- Use one VRS chain to implement one service level.

  For example, you can assign through the ACS routines the same management class (CATALOG) to all the data sets that you want to retain while they are cataloged. Then you can add a data set name VRS for this MC as shown in Figure 10-27.

Example 10-27 Define a new management class VRS

```
RMM ADDVRS DSNAME('CATALOG') WHILECATALOG
```

The rest of the VRSs with the same specification can be removed.

- Code MC ACS routine to process allocations for system-managed tapes and for non-system-managed tapes.

  All the data set specifications, system-managed or non-system-managed, tape data sets are to be managed in the same way, using the RMMVRS environment call for non-SMS tapes and regular allocations for SMS tapes.

- Assign policies by name. Move decisions out of EDGUX100.

  ACS routines are easier to change and maintain than exits, so it is recommended to include all data set specifications in the ACS routines, and remove them from EDGUX100 exit.

  The migration from MV to MC is transparent. The MC specification overwrites MV specification (refer to Figure 10-17 on page 408) and the data sets are retained by policy, for example:

  a. We create the VRS as shown in Figure 10-28.

Example 10-28 ADD a VRS management class definition

```
RMM ADDVRS DSNAME('CYCL005') CYCLES COUNT(5)
```

  b. We create the data sets 1, 2, and 3 as shown in Figure 10-39 with a management value of CYCL005 assigned by EDGUX100.

Archived

c. We add the same specification in the EDGUX100 to ACS routines, assigning to this data sets a MC=CYCL005, and we create the data sets 4, 5 and 6 in Figure 10-39. They have MC=CYCL005 and MV=CYCL005.

d. After that, we remove the specification in EDGUX100, and we create the data set 7 in Figure 10-39. This data set has only MC=CYCL005.

e. DFSMSrmm retains five versions of the data sets: 7, 6, 5, 4 and 3, although numbers 1 and 2 have only MV specification.

```
1. Data set name YCJRES.TEST1 creation date 33 days ago with MV
2. Data set name YCJRES.TEST1 creation date 32 days ago with MV
3. Data set name YCJRES.TEST1 creation date 31 days ago with MV
4. Data set name YCJRES.TEST1 creation date 29 days ago with MV and MC
5. Data set name YCJRES.TEST1 creation date 28 days ago with MV and MC
6. Data set name YCJRES.TEST1 creation date 25 days ago with MV and MC
7. Data set name YCJRES.TEST1 creation date 20 days ago with MC
```

Figure 10-39 Data set examples

- Move entries in UXTABLE to ACS routines.

  The EDGUX100 exit is shipped by DFSMSrmm as SAMPLIB member EDGCVRSX. This sample exit uses a dynamic table that is initially built by the sample job EDGJSRDS. This table is called UXTABLE, and it is a list of assembler macro statements that is easily customized to add or remove entries. You can add your pooling decisions to the table by modifying the entries built by EDGCSRDS, or by adding new entries.

  If you are using UXTABLE to assign policies to data sets, you can eliminate this table assigning a MC through ACS routines to each data set name or data set name mask with an entry in the UXTABLE.

Archive

11

Tailoring your DFSMSrmm environment

At this point you have successfully implemented DFSMSrmm, run it in production, and have removed the TLMS from your system. In this chapter we describe how to implement additional function, which is provided in DFSMSrmm or supports DFSMSrmm, and how to tailor DFSMSrmm to your environment.

Objectives of this chapter:

- Implement additional function that supports DFSMSrmm.
- Tailor DFSMSrmm function to your environment

At the end of this chapter you will have an understanding of additional function in DFSMSrmm and how to customize it for your environment. You will learn about products that enhance DFSMSrmm function.

© Copyright IBM Corp. 2004. All rights reserved.

11.1 IBM Tivoli Workload Scheduler

To manage the resources defined to DFSMSrmm, you need to run the inventory management tasks, also known as *housekeeping* periodically. DFSMSrmm provides the EDGHSKP utility to help you perform inventory management. A job scheduling product like IBM Tivoli Workload Scheduler for z/OS (which incorporates Tivoli Operations Planning and Control (OPC)) will be helpful to run these tasks.

IBM Tivoli Workload Scheduler for z/OS is a system management product that functions as a batch job scheduler. You can configure the typical DFSMSrmm batch job flow to IBM Tivoli Workload Scheduler for z/OS to run inventory management jobs regularly. This provides the following benefits:

- Prevents more than one EDGHSKP job from running at the same time
- Avoids jobs failing because inventory management is already running
- Prevents EDGUTIL VERIFY jobs from running at the same time as an inventory management jobs
- Handles the dependencies between jobs

The sample IBM Tivoli Workload Scheduler for z/OS stream included in the DFSMSrmm SAMPLIB library proposes the following steps when running inventory management. The name between ( ) is the sample member name included in SAMPLIB:

- Preparation (EDGJHKPA)
- CDS verify (EDGJVFY)
- CDS backup (EDGJBKP1, EDGJBKP2)
- Inventory management (EDGJDHKP(daily), EDGJWHKP(weekly))
- Erasing and labeling volumes (EDGJINER)
- Scratch processing (EDGJEXP)
- Scratch reporting (EDGJSCR)
- Ejecting volumes from a system-managed tape library (EDGJEJC)
- Producing reports (EDGJMOVE)
- Move confirmation (EDGJCMOV)

Also included in SAMPLIB is the IBM Tivoli Workload Scheduler for z/OS batch loader sample JCL, EDGJLOPC.

11.1.1 IBM Tivoli Workload Scheduler overview

This section introduces the basic IBM Tivoli Workload Scheduler for z/OS concepts, which you need to be familiar with for a better understanding of the samples provided by DFSMSrmm. Note that the Tivoli OPC product has been renamed to *IBM Tivoli Workload Scheduler for z/OS*.

The basic IBM Tivoli Workload Scheduler for z/OS components are:

- **Calendars** contain the information about the working days and free days for work planning purposes.
- **Workstations** are the logical place where work occurs. There are three types of workstations:
  - Computer workstations
  - Printer workstations
  - General workstations

  The majority of the IBM Tivoli Workload Scheduler for z/OS operations run in computer workstations. The activity that occurs at each workstation is known as *operation*. Each operation has a *job name*, *workstation ID* and *operation number*.
- **Applications** are units of production work that IBM Tivoli Workload Scheduler for z/OS schedules. Each application can be one or more operations.
- **Special resources (optional)** represents any type of limited resources like data sets, tape drives, or communication lines.

IBM Tivoli Workload Scheduler for z/OS can schedule jobs on a planned basis or dynamically using event-triggered scheduling. The *Event-Triggered Tracking* (ETT) gives you a method of controlling and tracking workload that cannot be planned in advance. Some of the situations that we can consider in this special scheduling are the journal full condition and low-on-scratch situation.

The DFSMSrmm batch loader sample helps you in the definition of applications and special resources. Calendars and workstations must be defined using the IBM Tivoli Workload Scheduler for z/OS dialog.

11.1.2 Schedule your inventory management tasks

The schedule of inventory management tasks should be defined before running. You can schedule each DFSMSrmm utility to run on its own, or schedule many activities to run together in a sequence. Some activities need to be performed daily like vital records processing, while other activities may be done less frequently.

Table 11-1 shows our recommendation. This may help you to decide how frequently to run DFSMSrmm utilities for your requirement.

Archived

Table 11-1 Schedule inventory management tasks

Activity	Frequency
Processing vital records daily	Daily
Performing expiration processing	Daily
Performing storage location management	Weekly
Creating an extract data set	Daily
Initializing and erasing volumes	Weekly
Creating security and audit reports	Monthly
Backing up the DFSMSrmm control data set	Daily
Backing up the DFSMSrmm journal	Daily
Verifying the control data set	Monthly

We can use the IBM Tivoli Workload Scheduler for z/OS batch loader sample JCL, EDGJLOPC to define the schedule for the activities shown in Table 11-1. The sample JCL defines the following applications based on the execution frequency:

- Daily applications:
  - RMMBKP
  - RMMHKPD
  - RMMPOST
  - RMMEXP
- Weekly applications:
  - RMMWEEK
  - RMMMOVE
- Monthly applications:
  - RMMMTH
- Other applications:
- RMMVRSVER is run only when a previous application finished with a RC=8

The flow of the IBM Tivoli Workload Scheduler for z/OS batch loader sample is shown in Figure 11-1.

Archive

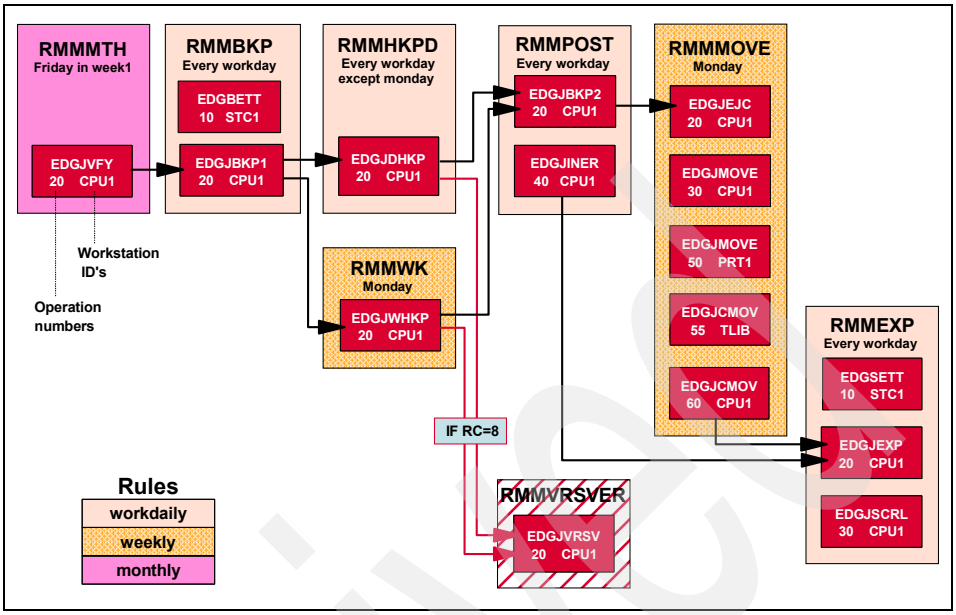


Figure 11-1 DFSMSrmm IBM Tivoli Workload Scheduler for z/OS job flow

In each application, you can see the different operations included. These are the jobs that perform each of the steps introduced at the beginning of the chapter. There are two more jobs: EDGBETT and EDGSETT that are defined under ETT. EDGBETT triggers RMMEXP application in case a low-on-scratch condition is reached, and EDGSETT triggers RMMBKP application in case the journal full threshold is reached.

When you have decided on your schedule, you should customize the EDGJLOPC sample to update the sample schedule to meet your needs. Refer to *DFSMSrmm Implementation and Customization Guide*, SC26-7405 for more details.

In the following sections, we cover the different activities that are performed in each of the steps suggested that comprise the whole inventory management process.

11.1.3 Preparation

This first step is performed by sample job EDGJHKPA. This job executes the EDGPHKPA procedure for allocating the inventory management files needed. Inside the EDGPHKPA procedure there are some others, each allocating one different file:

- EDGPMSGA: allocates the (+1) version of MESSAGE file
- EDGPVRSA: allocates the (+1) versions of REPORT and ACTIVITY files
- EDGPACTA: allocates the files needed for the ACTIVITY reports:
  - REPORT.RUNINFO
  - REPORT.VRS
  - REPORT.VRS.SUMMARY
  - REPORT.RETDATE
  - REPORT.RETDATE.SUMMARY
  - REPORT.MATCHVRS
  - REPORT.MATCHVRS.SUMMARY
  - REPORT.SUBCHAIN
  - REPORT.SUBCHAIN.SUMMARY
- EDGPRPTA: allocates the (+1) version of the REPTEXT file

This procedure also allocates the SCRDATE file.

11.1.4 CDS verify

A sample IBM Tivoli Workload Scheduler for z/OS job EDGJVFY is provided for verifying the CDS contents. The job invokes a EDGPVFY procedure, which uses the EDGUTIL program with PARM=VERIFY. The expected completion is the return code 0. You must perform manual recovery before dependent jobs can be run when the return code is greater than 0.

For more information about verifying the CDS, refer to 11.9.1, “Verify” on page 474.

11.1.5 CDS backup

Sample jobs EDGJBKP1 and EDGJBKP2 are provided in SYS1.SAMPLIB. These jobs are used to backup the CDS and journal, and to clear the journal after a successful backup. The jobs invoke the EDGPBKUP procedure, which executes EDGHSKP program with PARM=’BACKUP(*type*)’, *type being* one of DSS or AMS. If you select DSS, then you are able to use concurrent copy for backing up the CDS. It also allocates the files needed for the backups.

There is another procedure inside the jobs, EDGPMSGC, which allocates other files needed for the backup process.

For more information about backing-up the CDS, refer to 11.1.5, “CDS backup” on page 438.

Archived

11.1.6 Inventory management

DFSMSrmm manages the removable media resources and performs the inventory management (housekeeping) according to the policies defined to DFSMSrmm as vital record specifications (VRSs).

The inventory management processing includes the following tasks:

- VR selection (VRSEL)
  The purpose of vital records processing is to identify the volumes that should be retained and how they should be moved based on the vital record specifications you defined.
  Refer to Chapter 10, “Policy management” on page 391 for detailed information about how VRSs work.
- Expiration processing (EXPROC)
  This processing is used to expire volumes to scratch status according to the management policy. DFSMSrmm calls the EDGUX200 installation exit when running the expiration processing. This exit is called every time a volume is identified as ready to return to scratch. During expiration processing, DFSMSrmm:
  - Checks the expiration date to identify volumes that are ready to expire
  - Determines the type of release action needed for the volumes that have reached their expiration date
  - Manages the release actions for volumes before returning them to the scratch pool or removing them from the library
  - Removes the catalog entry and uncatalogs all parts of the data set across all volumes in the set for a multivolume data set
  - Optionally, updates RACF tape security profiles for volumes returning to scratch status
  - If UNCATALOG(Y or S) is specified in PARMLIB member EDGRMMxx, DFSMSrmm will uncatalog all data sets for volumes returning to scratch status.
  - When the EDGRMMxx PARMLIB option RETAINBY(SET) is in use, DFSMSrmm ensures all volumes in a set are not expired if any one volume does not expire.
- Storage location management (DSTORE)
  Requests storage location management to set the destination for the volume and optionally assigns the exact shelf location to be used. Storage location management processing assigns volume destinations and sets required volume destinations based on the results of other inventory management functions.

Archive

You can select to perform DSTORE by location, specifying the EDGHSKP parameter as shown in Example 11-1.

Example 11-1 EDGHSKP parameter DSTORE

```
DSTORE(LOCATION(from_location:to_location,...)
```

where:

`from_location:to_location` Is a pair of location names separated by a colon. The `from_location` is the current location a volume should move from.

`to_location` Is the name of the required location where a volume should move to. If you omit to_location, DFSMSrmm uses "*" as the default.

You can specify one to eight pairs of *`from_location:to_location`* names. DSTORE processing will be performed then for a volume, if it matches to one of the location pairs specified.

The from_location and to_location names can be specified in one of the following ways:

Specific A specific location name is 1 to 8 characters. The location names you specify are not validated against the DFSMSrmm LOCDEF entries or the name of SMS libraries.

Example 1: SHELF. Means just this one location is included

Generic A generic location name is 1 to 7 characters followed by any generic characters valid in a single data set qualifier.

Example 2: ATL*. Means all location names starting with the characters ATL are included

Example 3: *. Means all locations are included

Example 4: ABC%%%. Means all locations starting with ABC and with any other character up to a total of a 6 character name.

- Report extract (RPTEXT)
  When running inventory management, you can choose to request an extract data set, which can be used to create various reports such as volume movement or inventory.
- Trial run (VERIFY)
  Trial run vital record processing does not change data set and volume information in the DFSMSrmm control data set. Use trial run vital record processing to analyze the effect that your movement and retention policies will have.

To run inventory management functions, sampleIBM Tivoli Workload Scheduler for z/OS jobs EDGJDHKP and EDGJWHKP are provided in SYS1.SAMPLIB.

- EDGJDHKP - daily job. It invokes the following procedures:
  - EDGPHKP: runs EDGHSKP program with PARM='VRSEL,EXPROC,RPTEXT'.
  - EDGPMSGC, EDGPVRSA, and EDGPRPTA: for allocating other needed files.
- EDGJWHKP - weekly job. It invokes the following procedures:
  - EDGPHKP: runs EDGHSKP program with PARM='DSTORE,VRSEL,EXPROC,RPTEXT'.
  - EDGPMSGC, EDGPVRSA, and EDGPRPTA: for allocating other needed files.

11.1.7 Erasing and labeling volumes

This step automates the labelling and erasing of tape volumes. Sample IBM Tivoli Workload Scheduler for z/OS job EDGJINERS are provided in SYS1.SAMPLIB.

DFSMSrmm search for volumes with INITIALIZE or ERASE actions pending and labels them after optionally erasing them.

The job invokes the EDGPINER procedure, which runs the EDGINERS utility with the required parameters.

Refer to 11.4, "Initialize and erase volumes" on page 463 for detailed information on erasing and initializing tape volumes.

11.1.8 Scratch processing

This step is needed to process volumes that are returning to scratch status.

A sample IBM Tivoli Workload Scheduler for z/OS job is available in SYS1.SAMPLIB member EDGJEXP. This job invokes the EDGPEXP procedure, which runs EDGHSKP with the EXPROC parameter to perform expiration processing. It also allocates some other needed files.

For more information about the expiration processing performed as part of the (EXPROC), refer to 11.1.6, "Inventory management" on page 439.

Archived

11.1.9 Scratch reporting

Perform this step when you need to obtain information about volumes in scratch status. Sample job EDGJSCRL is provided in SYS1.SAMPLIB.

This job invokes the following procedures:

- EDGPRPTX: Creates an updated DFSMSrmm report extract file. For doing this, it runs the EDGHSKP utility with RPTEXT parameter. It allocates the new report extract file using the procedure EDGPRPTA.
- EDGPSCRL: It uses EDGRPTD reporting utility for generating the latest scratch lists using files SCRLIST and NEWSCR.

11.1.10 Ejecting volumes from a system-managed tape library

This task ejects volumes from system-managed tape libraries when volume movement is required. You should also consider VTS export processing in this step. Similar processing is required for any non-system-managed volumes, which you have in your library.

A sample IBM Tivoli Workload Scheduler for z/OS job is available in SYS1.SAMPLIB member EDGJEJC. This job invokes EDGPEJC procedure. You must execute this procedure once for each library in your installation, changing the library name (FROM), destination (TO), and output station (EJECT) in each case.

EDGPEJC procedure has two steps:

1. Creates a list of volumes to be moved from a system-managed library. The list is created using the command shown in Example 11-2.

Example 11-2 TSO RMM search volume command

```
RMM SV VOL(*) OWN(*) LIMIT(*) LOC(&FROM) DEST(&TO) INTRANSIT(N) -
    CLIST("RMM CV","EJECT(&EJECT)")
```

   The parameter CLIST allows you to create a set of commands for all the volumes in the generated list.

2. Uses the set of commands generated in the previous step for ejecting the volumes. The general command is shown in Example 11-3.

Example 11-3 TSO RMM change volume command

```
RMM CV volser EJECT(&EJECT)
```

11.1.11 Producing reports

When running inventory management, you can choose to request an extract data set that can be used to create various reports such as volume movement or inventory.

The extract data set contains information from the control data set. You can use the extract data set as input to EDGRPTD reporting utility, the EDGRRPTE exec, and the report generator to create reports. The extract data set may contain extended extract records.

See *DFSMSrmm Reporting* for information about using EDGRPTD and EDGRRPTE.

This step is used to generate movement reports. A sample IBM Tivoli Workload Scheduler for z/OS job is available in SYS1.SAMPLIB member EDGJMOVE. It invokes the following procedures:

- EDGPRPTX: Creates an updated report extract file, running EDGHSKP utility with RPTEXT parameter.
- EDGPMOVE: Uses the updated report extract file as input to EDGRPTD reporting utility to create movement reports. The produced reports are:

  RDYTOSCR Volumes to be moved from locations to home location
  TOSTRCK Report sorted by rack number including volumes to be moved:
  - From SHELF to storage locations
  - Between system-managed libraries
  - From system-managed libraries to storage locations

  TOSTOWN Report sorted by owner including volumes to be moved:
  - From SHELF to storage locations
  - Between system-managed libraries
  - From system-managed libraries to storage locations

  FMSTBIN Report sorted by bin number including volumes to be moved:
  - From storage location to SHELF
  - Between storage locations
  - From storage locations to system-managed libraries

  FMSTOWN Report sorted by owner including volumes to be moved:
  - From storage location to SHELF
  - Between storage locations
  - From storage locations to system-managed libraries

11.1.12 Move confirmation

Confirmation is needed for volume movement to locations outside the tape library.

Archived

After the inventory management processing such as vital record processing and the storage location management processing, DFSMSrmm needs your confirmation about volume movement.

A sample IBM Tivoli Workload Scheduler for z/OS job EDGJCMOV is provided in SAMPLIB for performing the move confirmation actions. A global confirmation command is issued, which means that the confirmation is for multiple volumes. The actual confirmation of movement for each volume is done in the next inventory management run.

The procedure EDGPCMOV is invoked, which issues the global confirm move command as shown in Example 11-4.

Example 11-4 TSO RMM confirm move command

```
RMM CV * CMOVE(&FROM,&TO,&TYPE)
```

You should provide location names valid for your installation in the parameters FROM, TO, and TYPE, or use ALL to confirm all outstanding moves.

11.1.13 Additional considerations

There are some additional considerations to take into account for inventory management related tasks.

Managing tape with permanent errors

DFSMSrmm tracks the permanent I/O errors for a volume, and marks it as it needs to be replaced if it is set with a return to scratch release action. DFSMSrmm does not identify those volumes that need the “return to owner” release action.

You can use the RMM **SEARCHVOLUME** subcommand, the extract data set or the DFSMSrmm ISPF dialog to check for volumes with temporary errors.

Managing open data sets

In the MESSAGE file, DFSMSrmm lists tape data sets that might be open as well as messages describing the expiration processing.

These open data sets might be produced by jobs that are currently active. However, some of them also might result from a job failure, meaning the data set was never closed. Review this list and make the necessary corrections. During vital record processing, DFSMSrmm matches open data sets to vital record specifications using the reserved data set name mask OPEN, or to vital record specifications defined with data set name masks.

Other considerations

You can perform some activities only under specific conditions:

- If you use system-managed tape library in your environment, and there are multiple TCDBs, you must run inventory management expiration processing at least once against each TCDB.
- When you want to retain data sets based on the catalog information, or if you specify the UNCATALOG with Y or S in EDGRMMxx to indicate that catalog processing is required, you must run inventory management on a system that has access to the catalogs, or you have to enable catalog synchronization.
- In an unshared user catalog environment, specify **OPTION** command with CATSYSID operand in the EDGRMMxx to use DFSMSrmm. You must ensure CDS and user catalogs are synchronized and run expiration processing on each different catalog environment.
- The DFSMSrmm subsystem address space performs all inventory management functions, except backing up the control data set. It is EDGHSKP that backs up the DFSMSrmm control data set.
- You can continue using RMM TSO subcommands and DFSMSrmm continues to record tape usage, while inventory management functions are running.
- At least one vital record specification must be defined in order to run DFSMSrmm vital record processing. The EDGRMMxx PARMLIB OPTION command VRSMIN operand provides control over this processing.

11.2 Activate RACF TAPEVOL and TAPEDSN

You can control the actions that DFSMSrmm takes on RACF tape profiles through the DFSMSrmm OPTION TPRACF command and your installation's RACF options. To protect tape volumes, you can specify DFSMSrmm OPTION TPRACF(AUTOMATIC) or TPRACF(PREDEFINED). See 11.2.4, "DFSMSrmm OPTION command operand TPRACF" on page 447 for more information.

Example 11-5 Activate tape data set protection

```
SETROPTS CLASSACT(TAPEVOL)
SETROPTS TAPEDSN
SETROPTS LIST
```

Before you execute the RACF commands, activate RACF TAPEVOL and TAPEDSN classes as shown in Example 11-5.

Recommendation: We recommend that you do not activate the RACF TAPEVOL and TAPEDSN classes until after you have converted to and successfully used DFSMSrmm for a period of time (for example, three months).

For optimal security, we recommend to make TAPEVOL and TAPEDSN active with either EDGRMMxx PARMLIB option TPRACF(P) or TPRACF(A) to obtain full protection at the volume and data set level.

Although we discourage this, it is possible to have no tape protection, or to use only TAPEVOL profiles. If you have no tape security, you cannot control the creation and use of data on tape. If you use only TAPEVOL profiles to protect tape volumes, you must maintain the access lists to the TAPEVOL profiles to allow access. Consider the use of TAPEDSN, so that access to tape data is covered by your normal, existing, generic data set profiles.

11.2.1 Before you can start

You can implementing RACF TAPEVOL and TAPEDSN only if you have a user with all required authority and access.

Your user ID needs the following attributes to execute all created RACF commands:

SPECIAL Specifies that the user is to be allowed to issue all RACF commands with all operands except the operands that requires the AUDITOR attribute

AUDITOR (Optional) Specifies that the user is to have full responsibility for auditing the use of system resources. An AUDITOR user can control logging of detected accesses to any RACF-protected resources during RACF authorization checking and access to the RACF data set.

You need access to following data set profiles and resources in class FACILITY defined in RACF to create a new report extract data set:

- ALTER access to data set profile RMM.HSKP.**
- UPDATE access to data set profile SYS1.PARMLIB or any equivalent PARMLIB you have defined EDGRMMxx member.
- READ access to STGADMIN.EDG.HOUSEKEEP defined in FACILITY class
- READ access to STGADMIN.EDG.HOUSEKEEP.RPTEXT defined in FACILITY class

11.2.2 DFSMSrmm RACF tape security support

The objective for DFSMSrmm RACF tape security support is to provide a complete interface with RACF for tapes, so you can use any combination of valid RACF options, including use of any RACF installation exits you still have. You can tell DFSMSrmm not to provide any RACF support by specifying the DFSMSrmm EDGRMMxx PARMLIB OPTION TPRACF(N) command. You can set up the type of processing wanted by requesting automatic TPRACF(A) or predefined TPRACF(P) RACF profiles. You can also use the DFSMSrmm EDGRMMxx PARMLIB VLPOOL RACF command to provide control of RACF at the pool level.

11.2.3 DFSMSrmm automatic tape security support

DFSMSrmm automatic tape security processing assumes that when a tape is mounted for output as a scratch tape, it will not be RACF protected. During the open processing for the tape data set either DFSMSdfp or your installation exits will cause some RACF profiles to be created. DFSMSrmm predefined processing also ensures that when TAPEVOL and TAPEDSN are active, RACF predefined TAPEVOL profiles with an empty TVTOC are created for scratch volumes. To enforce a DFSMSrmm standard that all tapes are RACF protected when the data set is closed, DFSMSrmm checks that a RACF security profile exists for the volume. If a RACF security profile does not exist, DFSMSrmm creates one and places the owner of the tape in the access list for the TAPEVOL profile with ALTER authority. The process ensures that your current tape security mechanism should continue to work as long as it is based on TAPEVOL profiles.

11.2.4 DFSMSrmm OPTION command operand TPRACF

This specifies the type of RACF tape support that DFSMSrmm provides. Use this operand when you want DFSMSrmm to maintain the security profiles that protect tape volumes. RACF tape support you define can be overridden by RACF tape support you define with the VLPOOL described in 11.2.5, “DFSMSrmm VLPOOL command operand RACF” on page 448.

The TPRACF(NO) option is assumed for any volume serial number containing special characters. To protect tape volumes that use special characters in the volume serial number, use RACF generic TAPEVOL profiles that are outside of DFSMSrmm control. DFSMSrmm honors all other TPRACF options for volume serial numbers that are alphanumeric including national characters. You can also use RACF generic data set profiles to protect data created on the tape volume. DFSMSrmm honors the TPRACF setting when running in all modes.

If you set TPRACF(PREDEFINED) or TPRACF(AUTOMATIC), DFSMSrmm ensures that all non scratch tapes are protected by a discrete RACF TAPEVOL profile by checking that a RACF profile exists, whenever a data set is written on a

tape. If a profile does not exist, DFSMSrmm creates one. Therefore, you do not need to use RACF installation exits to set the JCL PROTECT=YES option or specify PROTECT=YES in your JCL. You can use generic data set profiles for all tape data sets without changes to JCL or installation procedures, if you used the VLPOOL command with the RACF(Y) operand, because DFSMSrmm creates a TVTOC when you use the RACF TAPEDSN option.

For both TPRACF(PREDEFINED) and TPRACF(AUTOMATIC), DFSMSrmm ensures TAPEVOL profiles are always deleted during recycling of scratch volumes, if they exist. When you use a volume for output and there is no RACF protection for the volume when you close the data set, DFSMSrmm creates a profile to protect the volume. The owner of the volume is given RACF access to the profile. If TAPEVOL and TAPEDSN are active, DFSMSrmm creates a TVTOC and adds the first file data set entry.

Default: TPRACF(N)

For more information on RACF processing and security options, refer to the manual *DFSMSrmm Implementation and Customization Guide,* SC26-7405*,* and chapter titled “Authorizing DFSMSrmm Users and Ensuring Security.”

11.2.5 DFSMSrmm VLPOOL command operand RACF

Specifies the type of RACF tape support that DFSMSrmm should provide for volumes in the pool you are defining. When you are defining RACF tape support for volumes in a pool, you must look at the RACF tape support you have defined for your installation with the OPTION TPRACF command described in the manual *DFSMSrmm Implementation and Customization Guide,* SG26-7405, and the chapter titled “Defining System Options: OPTION.” Specify **Y** if you want DFSMSrmm to create RACF tape profiles for the volumes in the pool. Ensure that **OPTION TPRACF(AUTOMATIC or PREDEFINED)** is specified. If **OPTION TPRACF(NO)** is specified, DFSMSrmm will not create the RACF tape profiles.

Only specify `Y` for tape pools because TAPEVOL profiles are not required for optical volumes.

Specify `N` if you do not want DFSMSrmm to create RACF tape profiles for the volumes in the pool. If you specify `N`, DFSMSrmm plays no part in creating or deleting RACF tape profiles, regardless of the system-wide option TPRACF.

Default: RACF(N)

11.2.6 Running DFSMShsm and DFSMSrmm

How you use the DFSMShsm SETSYS option TAPESECURITY influences the TPRACF value in the DFSMSrmm PARMLIB member EDGRMMxx. DFSMShsm uses RACF TAPEVOL profiles to secure its volumes. You can combine the DFSMShsm method for securing volumes with DFSMSrmm RACF options as described in *DFSMSrmm Implementation and Customization Guide*, SG26-7405, and the chapter titled “Defining System Options: OPTION” to ensure complete security for your tape data.

Select the appropriate combinations for your environment from the following options:

- Use RACF with TAPEVOL and TAPEDSN.
- Use DFSMShsm TAPE SECURITY with RACF or EXPIRATIONINCLUDE.
- Use DFSMSrmm TPRACF with A, P, or N and VLPOOL RACF with Y or N.

Recommendations:

- Run DFSMShsm with scratch tapes that DFSMSrmm manages so you can have a single scratch pool for all users of tape, and so you can gain any benefits available from pre-mounting of scratch tapes in cartridge loaders.
- Use RACF tape security to ensure data security on your system.
- Set the following suggested DFSMShsm parameters:

```
SETSYS EXITOFF(ARCTVEXT)-
       SELECTVOLUME(SCRATCH)-
       TAPEDELETION(SCRATCH)-
       TAPESECURITY(RACF EXPIRATION)-
       PARTIALTAPE(MARKFULL)
```

  Specify EXITOFF(ARCTVEXT) if DFSMSrmm is your only tape management product because DFSMShsm always calls the DFSMSrmm programming interface EDGTVEXT if you have installed DFSMSrmm Release 1.4 or higher.

11.2.7 Define RACF profiles

Example 11-6 shows a sample batch job to protect tape volumes and tape data sets using RACF RDEFINE, RALTER, and ADDSD commands.

Example 11-6 RACF commands for TAPEVOL and TAPEDSN implementation

```
/* ************************************************************ */
/* RACF cmds for DFSMSrmm TAPEVOL and/or TAPEDSN implementation */
/*                                                              */
/* ************************************************************ */
```

```
RDEFINE TAPEVOL HSMHSM OWNER(DFHSM) +
    UACC(NONE) AUDIT(ALL(READ))                        /* VOL-SET */
    PERMIT HSMHSM CLASS(TAPEVOL) ID(TRESTER) DELETE
RALTER TAPEVOL SC0239 TVTOC                             /* MASTER */
ADDSD 'BSYPPS.OPD.STAIR.TSTAIR.E9201' +
    FILESEQ(1) SETONLY TAPE UNIT(TAPE) VOL(SC0239)
ADDSD 'BSYPPS.OPD.STAIR.DSTAIR.E9201' +
    FILESEQ(2) SETONLY TAPE UNIT(TAPE) VOL(SC0239)
ADDSD 'BSYPPS.OPD.STAIR.DATASETS.E9201' +
    FILESEQ(3) SETONLY TAPE UNIT(TAPE) VOL(SC0239)
PERMIT SC0239 CLASS(TAPEVOL) ID(TRESTER) DELETE
RALTER TAPEVOL HSMHSM ADDVOL(SC0648)                     /* HBAC */
RALTER TAPEVOL SC0010 TVTOC                             /* MASTER */
ADDSD 'SCHLUM.RMMDEMO.FILE2.VOL12' +
    FILESEQ(1) SETONLY TAPE UNIT(TAPE) VOL(SC0010)
PERMIT SC0239 CLASS(TAPEVOL) ID(TRESTER) DELETE
```

Description:

- RDEFINE TAPEVOL HSMHSM
  DFSMShsm protects tapes with RACF by adding them to a RACF tape-volume set. All tapes in DFSMShsm's RACF tape-volume set share the same access list and auditing controls.
- RALTER TAPEVOL SC0239 TVTOC
  Create a TVTOC for an existing TAPEVOL profile.
- ADDSD 'BSYPPS.OPD.STAIR.TSTAIR.E9201
  Add tape data set profiles to the TVTOC for volumes with DFSMSrmm status of MASTER.
- PERMIT SC0239 CLASS(TAPEVOL) ID(TRESTER) DELETE
  Remove the command executor from the access list.

11.2.8 Final RACF activities

After you have defined the DFSMSrmm RACF function groups, data set profiles, and resources in RACF FACILITY class, and assigned DFSMSrmm a user ID, you must refresh the RACF in-storage profiles.

Example 11-7 shows a batch job to REFRESH the RACF in-storage profiles.

Example 11-7 JCL to refresh RACF in-storage-profiles

```
//RACFREFR JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ****************************************************************** *
//*  Caution
//*  =======
//*    To execute this job you need the RACF "SPECIAL" attribute
```

```
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /*   FINAL ACTIVITIES                                             */
 /* ************************************************************** */
   SETROPTS GENERIC(FACILITY) REFRESH
   SETROPTS RACLIST(FACILITY) REFRESH
   SETROPTS GENERIC(DATASET) REFRESH
   SETROPTS GENERIC(STARTED) REFRESH
   SETROPTS LIST
//
```

11.3 Maintaining your control data set (CDS)

DFSMSrmm provides you tree different utilities to maintain your DFSMSrmm control data set (CDS). The CDS is a VSAM key sequenced data set (KSDS) and should be backed up once a day and reorganized from time to time. In this section, we describe how you can back up your CDS and journal, reorganizing your CDS, and restore your CDS. The utilities provide some more functions, but these functions are not described in this section. For more information about these utilities, refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

The following are the functions of the DFSMSrmm utilities to maintain the CDS that we describe in this section:

EDGHSKP	To back up the CDS and journal
EDGBKUP	To back up the CDS and journal, reorganize the CDS, or to restore the CDS
EDGUTIL	To create or update the CDS control record

11.3.1 Using EDGHSKP

You can use this utility to back up your DFSMSrmm CDS and journal, and to clear the journal. You do not have to stop any DFSMSrmm function while this utility is running. DFSMSrmm will stop updates made to the control data set while the AMS backup function is running. If you are using DSS to back up the CDS, there are updates allowed.

The EDGHSKP utility requests that DFSMSrmm is initialized and the task is started.

BACKUP

Specify BACKUP to back up the control data set to back up the journal, or to back up both the control data set and the journal. When you specify BACKUP on the EXEC statement, you must also specify a DD statement. Specify BACKUP DD when you back up the control data set, and JRNLBKUP DD when you back up the journal. You can also specify both DD statements in your EDGHSKP JCL.

We recommend the use of DFSMSrmm utilities to backup the CDS and journal data set instead of other backup tools. You can plan batch job using EDGHSKP or EDGBKUP for backup, or set up an automatic backup procedure as well.

You can run DFSMSrmm control data set backup processing using the DFSMSdss **DUMP** command, or the access method services **REPRO** command.

When you use DFSMSdss concurrent copy for backup, DFSMSrmm continues to process commands and update the control data set.

When you use the access method services **REPRO** command or DFSMSdss without concurrent copy for backup, DFSMSrmm does not allow updates to the control data set during control data set backup.

Certain DFSMSrmm requests wait until the backup completes. RMM TSO subcommands that request to update the DFSMSrmm control data set fail. Only actions that read the control data set continue uninterrupted.

To use DFSMSdss concurrent copy, you must authorize the DFSMSrmm batch inventory management backup job to the RACF profile STGADMIN.ADR.DUMP.CNCURRNT:

- **BACKUP(AMS)**
  BACKUP(AMS) is the default. DFSMSrmm uses the access method services REPRO command to perform backup processing. Specify BACKUP(AMS) to prevent DFSMSrmm from updating the control data set during control data set backup processing. Backup cannot be directly to tape.
- **BACKUP(DSS)**
  Specify BACKUP(DSS) with the DFSMSdss concurrent copy environment set up to permit the update of the DFSMSrmm control data set during backup processing. This ensures that all tape activities can continue during backup processing. You can specify BACKUP(DSS) without setting up the concurrent copy environment. If you specify BACKUP(DSS), backup can be direct to tape.

For journal backup, DFSMSrmm uses IDCAMS to back up the journal even when BACKUP(DSS) is specified. When using EDGHSKP to perform back up, the journal can be cleared.

Archived

Note: Do not specify the data set names in the JCL. If you do, the job fails. DFSMSrmm obtains the name of the control data set and journal from the running DFSMSrmm subsystem.

Example 11-8 shows you a sample EDGHSKP JCL using access method services to perform backup processing. The back up copies must be allocated on DASD.

Example 11-8 CDS and journal backing up using AMS

```
//CDSHKUP  JOB ....
//STEP0001 EXEC PGM=EDGHSKP,PARM='BACKUP'
//MESSAGE  DD   DISP=SHR,DSN=RMM.MESSAGE.FILE
//SYSPRINT DD   SYSOUT=*
//BACKUP   DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.BACKUP(+1)
//JRNLBKUP DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.JRNLBKUP(+1)
```

Note: When you using AMS, the back up copy must be allocated on DISK, because until the backup processes the DFSMSrmm, CDS is locked and no open to a tape data set is possible.

where:

- BACKUP DD if you want to back up the control data set.
- JRNLBKUP DD if you want to back up the journal.
- MESSAGE DD lists the messages the DFSMSrmm subsystem issues.
- SYSPRINT DD contains the utility program messages.

If you have specified both DD statements to back up the CDS and journal, and the function ended without any errors, the journal will be cleared at the end.

Example 11-9 shows you a sample EDGHSKP JCL using DFSMSdss to perform backup processing. In this case, the output data set can be allocated on TAPE, but in a case of recovery, the actual journal and the latest back up copy of the journal is needed.

Example 11-9 CDS and journal back up using DFSMSdss

```
//CDSHKUP  JOB ....
//STEP0001 EXEC PGM=EDGHSKP,PARM='BACKUP(DSS)
//MESSAGE  DD   DISP=SHR,DSN=RMM.MESSAGE.FILE
//SYSPRINT DD   SYSOUT=*
```


Archived

```
//BACKUP   DD   DISP=(,CATLG),UNIT=TAPE,
//         LABEL=(1,SL),
//         DSN=RMM.HSKP.BACKUP(+1)
//JRNLBKUP DD   DISP=(,CATLG),UNIT=TAPE,
//         LABEL=(2,SL),VOL=REF=*.BACKUP,
//         DSN=RMM.HSKP.JRNLBKUP(+1),
//         RECFM=VB,BLKSIZE=0,LRECL=9000
```

where:

- BACKUP DD if you want to back up the control data set.
- JRNLBKUP DD if you want to back up the journal.
- MESSAGE DD lists the messages the DFSMSrmm subsystem issues.
- SYSPRINT DD contains the utility program messages.

If you have specified both DD statements to back up the CDS and journal, and the function ended without any errors, the journal will be cleared at the end.

Optionally, you can specify a DSSOPT DD statement to customize the DFSMSdss DUMP command.

To set up automatic backup for CDS, follow these steps:

1. Set **OPTION** command in EDGRMMxx:
   - Specify the JOURNALFULL value. The default is 75%, and a zero value means disable the function.
   - Specify the name of backup procedure using the BACKUPPROC operand.
   - Create a backup procedure in PROCLIB.

   Refer to Figure 11-2 for the **OPTION** command in EDGRMMxx, and Figure 11-10 for a sample backup procedure.

```
OPTION
      ...
      BACKUPPROC(EDGCDSBK) -
      JOURNALFULL(75) -
      ...
```

Figure 11-2 PARMLIB setting for automatic backup

Example 11-10 Procedure for backing up CDS and journal

```
//EDGCDSBK PROC OPT=BACKUP(AMS)
//EDGHSKP  EXEC PGM=EDGHSKP,PARM='&OPT.'
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DISP=SHR,DSN=RMM.PROD.MESSAGE
```

Archived

```
//BACKUP   DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.BACKUP(+1)
//JRNLBKUP DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.JRNLBKUP(+1)
```

Note: When the inventory management task is running, the DFSMSrmm will not perform automatic backup. If there are multiple DFSMSrmm subsystems sharing CDS, running automatic backup on only one system is better, or define different JOURNALFULL values.

When the inventory management task is running, the DFSMSrmm will not perform automatic backup. If there are multiple DFSMSrmm subsystems sharing CDS, running automatic backup on only one system is better.

11.3.2 Using EDGBKUP

You can use this utility to reorganize and restore your DFSMSrmm CDS. Also, you can use this utility to back up your DFSMSrmm CDS and journal, and to clear the journal. You have to stop DFSMSrmm or use the QUIESCE command to quiescing the DFSMSrmm subsystem interface. For more information about QUIESCE, refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

The EDGBKUP utility can be used if DFSMSrmm is stopped or quiesced.

BACKUP

Specify BACKUP to back up the control data set, to back up the journal, or to back up both the control data set and the journal. When you specify BACKUP on the EXEC statement, you must also specify a DD statement. Specify BACKUP DD when you back up the control data set, and JRNLBKUP DD when you back up the journal. You can also specify both DD statements in your EDGHSKP JCL.

- **BACKUP or BACKUP(NREORG)**
  BACKUP(AMS) is the default. DFSMSrmm uses the access method services REPRO command to perform backup processing.
- **BACKUP(DSS)**
  Specify BACKUP(DSS) with the DFSMSdss concurrent copy environment set up to permit the update of the DFSMSrmm control data set during backup processing.

For journal backup, DFSMSrmm uses IDCAMS to back up the journal even when BACKUP(DSS) is specified. When using EDGHSKP to perform back up, the journal can be cleared.

Example 11-11 shows you sample EDGBACK JCL. DFSMSrmm is stopped or quiesced using the access method services to perform backup processing.

Example 11-11 CDS and journal backing up using AMS

```
//CDSHKUP  JOB ....
//STEP0001 EXEC PGM=EDGBKUP,PARM='BACKUP'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//JOURNAL  DD   DISP=SHR,DSN=RMM.PROD.JRNL
//BACKUP   DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.BACKUP(+1)
//JRNLBKUP DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.JRNLBKUP(+1)
```

where:

- BACKUP DD if you want to back up the control data set.
- JOURNAL DD identifies the DFSMSrmm journal to be used during backup processing.
- JRNLBKUP DD if you want to back up the journal
- MASTER DD identifies the DFSMSrmm control data set.
- SYSPRINT DD contains the utility program messages.

If you have specified both DD statements to back up the CDS and journal, and the function ended without any errors, the journal will be cleared at the end.

Note: If DFSMSrmm is stopped or quiesced, you have to code the MASTER and JOURNAL DD if you have requested to back up this files.

REORG

Specify BACKUP(REORG) to reorganize the control data set during backup processing. DFSMSrmm uses IDCAMS to back up the control data set, to optionally back up the journal, and to restore the control data set from the backup.

The DFSMSrmm subsystem must be stopped or quiesced when you reorganize the active control data set. DFSMSrmm reorganizes the control data set by first backing up the control data set and optionally the journal. Then DFSMSrmm restores the control data set from the backup control data set.

Archived

Note: Do not use the BACKUP(DSS) parameter to reorganize the control data set.

Before you start the EDGBKUP utility to reorganize the control data set, you have to stop or quiesce the DFSMSrmm subsystem. In our example, we only quiesce the DFSMSrmm subsystem. If you use quiesce, you do not have to stop all your tape activities, nor do you have to switch offline your tape drives.

To quiesce DFSMSrmm, use the MVS modify command as shown in Example 11-12.

Example 11-12 Quiescing the DFSMSrmm subsystem interface

```
MODIFY DFRMM,QUIESCE
    or
F DFRMM,QUIESCE
```

Example 11-13 shows you a sample EDGBACK JCL that can be used to reorganize your DFSMSrmm control data set after you get the message that the DFSMSrmm subsystem is quiesced.

Example 11-13 CDS and journal backing up using AMS

```
//CDSHKUP  JOB ....
//STEP0001 EXEC PGM=EDGBKUP,PARM='BACKUP(REORG)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//JOURNAL  DD   DISP=SHR,DSN=RMM.PROD.JRNL
//BACKUP   DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.BACKUP(+1)
//JRNLBKUP DD   DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(CYL,(100,50),RLSE),
//         DSN=RMM.HSKP.JRNLBKUP(+1)
```

where:

- BACKUP DD if you want to back up the control data set
- JOURNAL DD identifies the DFSMSrmm journal to be used during backup processing.
- JRNLBKUP DD if you want to back up the journal
- MASTER DD identifies the DFSMSrmm control data set.
- SYSPRINT DD contains the utility program messages.

Archived

After you have checked the success of the reorganization, you have to restart the DFSMSrmm subsystem using a MVS modify command as shown in Example 11-14.

Example 11-14 Restart DFSMSrmm after it has been quiesced

```
MODIFY DFRMM,M=xx
```

where:

- M=xx specifies the PARMLIB member name that should be used.

Note: To restart DFSMSrmm after it has been quiesced, the operator must issue the command specified with a member name.

RESTORE

Specify RESTORE to restore the control data set from a backup copy. When you specify the BACKUP DD statement, DFSMSrmm restores the control data set. When you specify the JOURNAL DD statement, forward recovery is started. You can specify both the BACKUP DD statement and the JOURNAL DD statement in the same job step to restore, and forward recover the control data set from journal backups and the journal data set.

The DFSMSrmm subsystem must be stopped or quiesced when you restore the active control data set:

- **Restoring the control data set without forward recovery**
  You can restore your DFSMSrmm control data set without forward recovery. If you do this, you can lose data if the back up copy is not an actual one, and no changes are made since the backup copy was created.

  Example 11-15 shows you a sample EDGBACK JCL that can be used to restore your DFSMSrmm control data set without forward recovery.

Example 11-15 Restoring a control data set without forward recovery

```
//CDSHKUP  JOB ....
//STEP0001 EXEC PGM=EDGBKUP,PARM='BACKUP(REORG)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//BACKUP   DD   DISP=(,CATLG),DSN=RMM.HSKP.BACKUP(0)
```

Important: Use this example only if you have no need for a forward recovery.

- **Restoring and forward recovery the control data set**
  You can restoring your DFSMSrmm control data set and forward recovery the

DFSMSrmm control data set. This is the normal way to have the same information as before the crash.

Example 11-16 shows sample EDGBACK JCL that can be used to restore your DFSMSrmm CDS and forward recover it from the journal data set.

Example 11-16 Restoring a control data set with forward recovery

```
//CDSHKUP  JOB ....
//STEP0001 EXEC PGM=EDGBKUP,PARM='BACKUP(REORG)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//BACKUP   DD   DISP=(,CATLG),DSN=RMM.HSKP.BACKUP(0)
//JOURNAL  DD   DISP=(,CATLG),DSN=RMM.PROD.JRNL
```

Where:

- BACKUP DD specifies the newest back up copy of your DFSMSrmm control data set.
- JOURNAL DD specifies the actual DFSMSrmm journal.
- MASTER DD identifies the DFSMSrmm control data set.
- SYSPRINT DD contains the utility program messages.
- Example 11-17 shows sample EDGBACK JCL that can be used to restore your DFSMSrmm CDS and forward recover it using the latest backup copy should your DFSMSrmm CDS be destroyed or if it is unavailable.

Example 11-17 Restoring a CDS from older back up with forward recovery

```
//CDSREST  JOB ....
//STEP0001 EXEC PGM=EDGBKUP,PARM='BACKUP(REORG)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//BACKUP   DD   DISP=(,CATLG),DSN=RMM.HSKP.BACKUP(-1)
//JOURNAL  DD   DISP=(,CATLG),DSN=RMM.HSKP.JRNLBKUP(0)
//         DD   DISP=(,CATLG),DSN=RMM.PROD.JRNL
```

where:

- BACKUP DD specifies the back up copy -1of your DFSMSrmm control data set.
- JOURNAL DD specifies the newest back up copy of the DFSMSrmm journal and second the actual DFSMSrmm journal.
- MASTER DD identifies the DFSMSrmm control data set.
- SYSPRINT DD contains the utility program messages.

Archived

Attention: If you forward recover using multiple journal data sets, concatenate the journal data sets in the order in which changes were originally made.

- Example 11-18 shows sample EDGBACK JCL that can be used to restore your DFSMSrmm CDS from a backup copy -1 and forward recover it using the latest journal backup and the backup -1. This procedure can be used when the DFSMSrmm CDS is destroyed or not available.

Example 11-18 .Recovery using journal data sets backed up by BACKUP(AMS)

```
//CDSREST  JOB ....
//EDGBKUP  EXEC PGM=EDGBKUP,PARM='RESTORE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//BACKUP   DD   DISP=SHR,DSN=RMM.HSKP.BACKUP(-2)
//JOURNAL  DD   DISP=SHR,DSN=RMM.HSKP.JRNLBKUP(-1)
//         DD   DISP=SHR,DSN=RMM.HSKP.JRNLBKUP(0)
//         DD   DISP=SHR,DSN=RMM.PROD.JRNL
```

Where:

- BACKUP DD specifies the back up copy - 2 of your DFSMSrmm control data set.
- JOURNAL DD specifies the back up copy -1, the newest back up copy of the DFSMSrmm journal and third the actual DFSMSrmm journal.
- MASTER DD identifies the DFSMSrmm control data set.
- SYSPRINT DD contains the utility program messages.
- The situation is the same as in the previous example, but the journal backup is made using BACKUP(DSS), the temporary journal backup, and the later ones should be applied in the correct order. Example 11-19 shows a sample JCL.

Example 11-19 Recovery using journal data sets backed up by BACKUP(DSS)

```
//CDSREST  JOB ....
//EDGBKUP  EXEC PGM=EDGBKUP,PARM='RESTORE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//BACKUP   DD   DISP=SHR,DSN=RMM.HSKP.BACKUP(-2)
//JOURNAL  DD   DISP=SHR,DSN=RMM.HSKP.JRNLBKUP(-2)
//JOURNAL  DD   DISP=SHR,DSN=RMM.HSKP.JRNLBKUP(-1)
//         DD   DISP=SHR,DSN=RMM.HSKP.JRNLBKUP(0)
//         DD   DISP=SHR,DSN=RMM.PROD.JRNL
```

Archived

where:

- BACKUP DD specifies the back up copy - 2 of your DFSMSrmm control data set.
- JOURNAL DD specifies the back up copy -2, backup copy -1, the newest back up copy of the DFSMSrmm journal and the actual DFSMSrmm journal.
- MASTER DD identifies the DFSMSrmm control data set.
- SYSPRINT DD contains the utility program messages.
- If you did not perform forward recovery when you restored the CDS backup, you can perform the recovery with the restored CDS as shown in Example 11-20.

Example 11-20 Forward recovery of restored CDS

```
//CDSREST  JOB ....
//EDGBKUP  EXEC PGM=EDGBKUP,PARM='RESTORE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//JOURNAL  DD   DISP=SHR,DSN=RMM.PROD.JRNL
```

where:

- JOURNAL DD specifies the actual DFSMSrmm journal.
- MASTER DD identifies the DFSMSrmm control data set.
- SYSPRINT DD contains the utility program messages.

11.3.3 EDGUTIL

Use the EDGUTIL utility to create or update the control data set control record. To create or update the control record, the user of EDGUTIL must have UPDATE or higher RACF access to the DFSMSrmm control data set.

After DFSMSrmm is implemented, you have only a need to use this utility to update the control record if you are implementing new functions, or you must correct one of the counts stored in the control record. In this section we describe how you can use the utility to implement new functions.

CATSYNCH

Specifies whether or not the DFSMSrmm control data set and system catalogs are synchronized or not:

- **NO**
  CATSYNCH(NO) clears the last synchronization date and time. Specify CATSYNCH(NO) to force synchronization of the DFSMSrmm control data set and user catalogs the next time inventory management is run.

Archived

- **YES**
  CATSYNCH(YES) sets the last synchronization date and time to the current date and time.

  Note: Do not specify CATSYNCH(YES) if you have not run the EDGHSKP utility with the CATSYNCH parameter on each system to synchronize the DFSMSrmm control data set with the user catalogs.

EXTENDEDBIN

Enables DFSMSrmm extended bin support, which allows the reuse of bins at the start of a move:

- **YES**
  When extended bin support is enabled, DFSMSrmm records additional information in the volume and bin record while a volume is moving from or to a bin-managed storage location.

  Note: Do not enable extended bin support until you have made sure that the same level of code has been installed for all the DFSMSrmm systems that share a control data set.

  Once it is enabled, extended bin support cannot be disabled.

STACKEDVOLUME

Enables DFSMSrmm stacked volume support:

- **YES**
  When you enable stacked volume support, DFSMSrmm uses the stacked volume records to manage the movement of the volumes contained in the stacked volumes.

  Note: Prior to enabling stacked volume support, ensure all systems using the control data set are at the same release level.

  You cannot remove stacked volume support after it is enabled.

Example 11-21 shows you a sample EDGUTIL JCL that can be used to set the last synchronization date and time to the current date and time.

Example 11-21 Set last synchronization date and time

```
//CDSUPDT  JOB ....
//STEP0001 EXEC PGM=EDGUTIL,PARM='UPDATE'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DISP=SHR,DSN=RMM.PROD.CDS
//SYSIN    DD   *
```

```
 CONTROL CATSYNCH(YES)
/*
```

11.4 Initialize and erase volumes

EDGINERS is the DFSMSrmm utility for initializing and erasing volumes. You should use this utility as part of the daily inventory management tasks. You can run EDGINERS in manual or in automatic mode. The recommendation is to run in automatic mode unless you need to make any individual change to a volume.

You can use the LABEL procedure, which runs EDGINERS for initializing or erasing volumes manually. This procedure allows you initialize and erase any tape volume whether or not it is defined to DFSMSrmm. If you are initializing a volume not defined to DFSMSrmm, once it is initialized it will be automatically defined. The LABEL procedure is supplied as the EDGLABEL member in the SAMPLIB data set.

You should avoid the manual use of EDGINERS by defining all the volumes to DFSMSrmm previously with the RMM **ADDVOLUME** command, and setting the pending release action to INIT. The following command adds 100 new scratch volumes starting on A00001. These volumes need to be labeled before using as the INIT(Y) operand indicates. Example 11-22 shows how you can add the volumes using the TSO RMM subcommand.

Example 11-22 TSO RMM ADDVOLUME subcommand

```
RMM AV A00001 COUNT(100) STATUS(SCRATCH) INIT(Y)
```

Or, if you have the volumes defined to DFSMSrmm, but not with the pending release action INIT, you can change it with the command shown in Example 11-23.

Example 11-23 TSO RMM CHANGEVOLUME subcommand

```
RMM CV A00001 INIT(Y)
```

You can also add duplicate volumes to DFSMSrmm to be initialized, however, you can also initialize duplicate volumes using EDGINERS, which will automatically add the volume to DFSMSrmm. Example 11-24 shows how you can add duplicate volumes using the VOL1 parameter.

Example 11-24 TSO RMM ADDVOLUME subcommand using VOL1 parameter

```
RMM AV A00001 STATUS(SCRATCH) INIT(Y) VOL1(ABC001)
```

EDGINERS automatic processing is enabled when the INIT and ERASE options are specified in the control data set. When the volumes have the pending release action INIT or ERASE, then they can be processed automatically by DFSMSrmm. There are some required parameters in the EDGINERS JCL EXEC statement for being able to process any volume: COUNT, INITIALIZE, ERASE, MEDIANAME, POOL, or LOCATION.

When EDGINERS is running as an automatic process, it does not require any operator intervention.

For more information about EDGINERS, and a complete description of the parameters, refer to *DFSMSrmm Guide and Reference, SC26-7404*.

11.5 Disposition processing

DFSMSrmm disposition processing is optional and provides support for the following:

- Providing operators with information to assist them in tasks like moving a tape to a specific location
- Generating sticky labels
- Updating the location where a volume resides

DFSMSrmm disposition processing occurs at CLOSE and EOV for each volume. DFSMSrmm does not provide support for asynchronous processing in a separate job step. To implement this function, refer to *z/OS V1R3.0 DFSMSrmm Implementation and Customization Guide,* SC26-7405.

To produce labels with DFSMSrmm, you can use the default output, it is a WTO on route code 13, or the OUTPUT JCL statement in the DFSMSrmm started procedure to dynamically allocate a SYSOUT file for each label.

In Figure 11-3, you can see the default label format for a tape cartridge.

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

 jobname_     crdate____
              expdt_____
 den
```

Figure 11-3 Default label format for a tape cartridge

The following JCL, shown in Example 11-25, is a sample of this function using the default output for the label.

Example 11-25 Sample JCL to create sticky label

```
//YCJRES1  JOB MSGCLASS=X,NOTIFY=&SYSUID
//*
//STEP1    EXEC PGM=IEBGENER,REGION=4M
//SYSIN    DD   DUMMY
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=YCJRES1.DSN9,DISP=SHR
//SYSUT2   DD   DSN=YCJRES1.TEST.DISP,DISP=(,KEEP),
//    SPACE=(CYL,(2,2)),VOL=SER=TST108,UNIT=3590
//DISPDD DD *
OB90.TST108.0001.THIS IS A TEST
```

Running this sample JCL, we get the message shown in Example 11-4.

```
EDG4054I OB90.TST108.0001.THIS IS A TEST
```

Figure 11-4 EDG5054I message display

and the sticky label you can see in Figure 11-5.

```
YCJRES1.TEST.DISP


 YCJRES1     2001/052
             2001/055
      IDRC 00121  027951 FB

 TST108 0001 SL       OB90
```

Figure 11-5 Example label format

where:

```
dsname=     YCJRES1.TEST.DISP
jobname=    YCJRES1
crdate=     2001/052
expdt=      2001/055
comp=       IDRC
lrecl=      00121
blksize=    027951
recf=       FB
volser=     TST108
```

Archived

```
seqn=       0001
lab=        SL
devc=       0B90
```

11.6 Software products and foreign tapes

DFSMSrmm provides support for software products and foreign tapes.

To manage software tapes, you can define the product to DFSMSrmm, and DFSMSrmm then tracks the volume as a product volume.

To take advantage of product volume management, you need to set up some DFSMSrmm PARMLIB options, define racks numbers, define products to DFSMSrmm, and define the volumes associated with these products.

11.6.1 Defining software products

To define software products and the volumes associated with them, you can use DFSMSrmm ISPF panels or TSO RMM subcommands. In the examples, we have used TSO RMM subcommands. You need to follow the following steps:

1. If you want to have a separate range for product volumes, you need to define a new VLPOOL in the EDGRMMxx PARMLIB member. Example 11-26 shows a VLPOOL where the prefix for the product volumes is PP.

Example 11-26 Add a software product VLPOOL

```
VLPOOL DESC('software products') EXPDTCHECK(Y)       -
      MEDIANAME(CARTS) PREFIX(PP*) RACF(N) TYPE(R)
```

2. Restart DFSMSrmm with the new PARMLIB member.
3. Define some empty rack numbers in the pool as shown in Example 11-27.

Example 11-27 ADD RACK subcommand

```
RMM ADDRACK PP0001 COUNT(100)
```

4. Define the products you have or you plan to have, using the TSO RMM command as shown in Example 11-28.

Example 11-28 ADD a program product

```
RMM ADDPRODUCT 5695-DF1 NAME('DFSMS') OWNER(YCJRES1)     -
    LEVEL(V02R10M00) DESCRIPTION('DFSMS Version 2.10')
```

5. You can use the TSO RMM ADDVOLUME subcommand to add volumes. The default for LEVEL is V01R01M00. If your level is different, you must specify it. Example 11-29 shows you how you can assign volumes to a product.

Example 11-29 ADD volumes to a product

```
RMM ADDVOLUME DZ11CO POOL(PP*) MEDIANAME(CARTS) -
   FEATCD(1234) STATUS(MASTER) NUMBER(5695-DF1) LEVEL(V02R10M00)
```

Recommendation: If you use the DFSMSrmm panels to add the volumes, DFSMSrmm uses the following defaults:

Status:	master
Retention period:	90 days
Release action:	return to owner

11.6.2 Foreign tapes

DFSMSrmm can manage tapes that do not follow your naming convention. These volumes usually are those that are sent to you from another site.

DFSMSrmm manages foreign tapes no differently than any other tapes. It records data set names, provides VRS retention and movement controls, and has expiration release options to help you identify how to remove the foreign tapes from your library. The only volumes that cannot be defined to DFSMSrmm are those that are duplicates of volumes already defined to DFSMSrmm.

The way you define foreign tapes to DFSMSrmm is basically the same way that you use to define software product tapes.

The following is an example of how to define foreign tapes:

- Decide on a range of rack numbers to be used for foreign tapes that enter your installation. Consider whether you want to have a separate range for each type of media. If you have both tape reels and tape cartridges to manage as foreign tapes, and want each to have its own range of rack numbers, you need multiple pool definitions.
- Define a new VLPOOL in DFSMSrmm PARMLIB member. For example, if the prefix for your foreign volumes is FR, you need to add a statement as shown in Example 11-31.

Example 11-30 ADD a VLPOOL to manage foreign volumes

```
VLPOOL DESC('foreign tapes') EXPDTCHECK(Y) -
       MEDIANAME(CARTS) PREFIX(FR*) RACF(N) TYPE(R)
```

- Restart DFSMSrmm with the new PARMLIB member.

- Define some empty rack numbers in the pool by using a TSO RMM ADDRACK command as shown in Example 11-31.

Example 11-31 Add 100 new racks

```
RMM ADDRACK FR0001 COUNT(100)
```

In this example, volume BB0001 will be retained for 90 days and will then be released. The release action, RETURN, prevents the volume from going to the scratch pool. You can use it to identify volumes that should be removed from the library, and either returned to their owner or destroyed.

- To identify volumes waiting for RETURN release action, you can use the **SEARCHVOLUME** subcommand in Example 11-32.

Example 11-32 Search for volumes waiting for return release action

```
RMM SEARCHVOLUME VOL(*) OWNER(*) LIMIT(*) STATUS(RELEASE) -
    ACTION(RETURN)
```

Use the list generated to pull the volumes from the library and return them to their owner. Once the volume is removed from the library, you can confirm to DFSMSrmm that it has been removed. Use the command as shown in Example 11-33 to confirm the release action for volume BB000.

Example 11-33 Confirm a single volumes release action return

```
RMM CHANGEVOLUME DZ100B CRLSE(RETURN)
```

When the RETURN action is confirmed, the volume is deleted from the CDS, and the rack it occupied is now empty.

11.7 DFSMSrmm volume pools

A *pool* is a group of rack numbers or volumes that share a common prefix. In DFSMSrmm, there are two categories of pools: *rack* and *scratch*.

- A *rack pool* is shelf space that can be assigned to hold any volumes. Although you can add scratch volumes to these pools, you cannot normally use these volumes to satisfy non-specific mount requests. A rack pool cannot be used with the DFSMSrmm system-based scratch pooling. You can use rack pools for:
  - Volumes that are temporarily brought into the library but will be returned to the owner after a period of time
  - Customer, foreign tapes, and software product volumes

  - Scratch volumes for use with either DFSMSrmm exit-selected scratch pooling, or pools based on storage group.

- A *scratch pool* is shelf space assigned to hold volumes for use with the DFSMSrmm system-based scratch pooling. The volumes assigned to this shelf space can be used to satisfy scratch requests as long as the volumes are in scratch status. Once the volume has been written to, it becomes a volume with MASTER status, until the volume is returned to scratch status. The volume remains in the same DFSMSrmm system-based scratch pool in that it occupies the same shelf space regardless of status.

DFSMSrmm supports the following basic types of pooling:

- Pools of shelf space that are based on rack number prefixes. You can create different pools with different characteristics such as the media name and management criteria. You must use shelf space pools to store volumes that do not match your installation selected volume ranges.
- Pools of volumes that are based on the volume serial number prefix. These volumes do not have a rack number or the rack number matches the volume serial number. Each range identifies different characteristics like media name and management criteria.
- Scratch pools can be one or more pools of volumes. Scratch pools can be based on name, SMS storage group, prefix, or system.
- When you pool by storage group and use SMS ACS processing to assign a storage group to a tape data set, or the volume is in a system-managed manual tape library, DFSMSrmm ensures that a volume from the correct storage group is mounted. You can use storage group for scratch pooling for system-managed manual tape libraries and non-system managed tape.

11.7.1 Defining pools

You must define any pool you plan to use with DFSMSrmm in the PARMLIB member EDGRMMxx using the **VLPOOL** command.

You should define at least one pool using the **VLPOOL** command in the DFSMSrmm PARMLIB member. Use the pools to assign a media name for shelf space and volumes based on a prefix. Also, define a default pool to ensure that volumes and racks that do not match to a more specific **VLPOOL** are added with correct media name and pool name.

The operands you can use in the **VLPOOL** command are the following:

- **DESCRIPTION('pool description')**
  Describes the pool. Use any description from 1 to 32 characters.

- **EXPDTCHECK(Y|N|O)**
  Indicates the way to automate the processing of unexpired tape data sets at the pool level:
  - Use **N** if you do not want DFSMSrmm to check or validate the expiration date of tape data sets. This is the recommended value when using DFSMSrmm-managed scratch pools.
  - Use **Y** if you want to ensure that all unexpired tape data sets are never overwritten.
  - Use **O** if you do not want DFSMSrmm to take any action, but allows the operator or automation software to reply as necessary to any WTO (IEC507D). This is the default value.
- **MEDIANAME(media_name)**
  Indicates a name or media type for all volumes in this pool. Some samples are: CARTS, TAPE, LONG, SHORT, 3480, 3490. Any name is valid because DFSMSrmm does not check if it is defined to MVS.
- **NAME(pool_name)**
  Specify any name to identify the pool in the operator messages and tape drive displays. You must code the RACK(999) operand of **MNTMSG** command for modifying the messages adding the pool name.
- **PREFIX(nnnnn*)**
  Specifies a generic shelf location used in operator messages, RMM TSO commands, and the ISPF dialog

  DFSMSrmm uses the prefix value to assign a newly defined volume to a scratch pool.
- **RACF(Y | N)**
  Specifies the type of RACF support you want DFSMSrmm to provide for the volumes in the pool

  If you want to create RACF tape profiles, then specify Y and be sure that OPTION TPRACF(A | P) is specified.
- **SYSID(system_name**)
  This operand associates a scratch pool with a particular system. DFSMSrmm matches the value with the SYSID operand of the **OPTION** command.
  If the VLPOOL SYSID matches the OPTION SYSID then:
  - A volume from this pool can be used to satisfy a scratch mount request on this system.
  - A volume from any other pool where its SYSID also matches the OPTION SYSID can be used to satisfy a scratch request on this system.

    A volume from a pool that has no explicit VLPOOL SYSID can be used to satisfy a scratch request on a system if there are no other VLPOOL definitions that have a SYSID matching the OPTION SYSID. The SYSID

Archived

values are ignored when a storage group is selected by ACS processing and when a pool is set for a non-specific volume request by EDGUX100 processing. The system level pools can still be used as long as SMS ACS processing does not assign a storage group and the EDGUX100 exit does not set a specific pool to be use.

- **TYPE(S | R)**
  Specifies the type of pool. Use **R** for rack pools and **S** for scratch pools.

In Figure 11-6, you can see some sample pools definitions.

```
VLPOOL RACF(N) TYPE(S) EXPDTCHECK(0) MEDIANAME(CARTS) -
       DESCRIPTION('3590 POOL') PREFIX(TST*) NAME(GENERAL)

VLPOOL RACF(N) TYPE(S) EXPDTCHECK(0) MEDIANAME(CARTS) -
       DESCRIPTION('3490 POOL') PREFIX(HGT*) NAME(GENERAL)

VLPOOL RACF(N) TYPE(S) EXPDTCHECK(0) MEDIANAME(CARTS) -
       DESCRIPTION('3490 POOL') PREFIX(VVV*) NAME(GENERAL)

VLPOOL RACF(N) TYPE(S) EXPDTCHECK(0) MEDIANAME(CARTS) -
       DESCRIPTION('DEFAULT POOL') PREFIX(*) NAME(DEFAULT)

VLPOOL RACF(N) TYPE(R) EXPDTCHECK(Y) MEDIANAME(CARTS) -
       DESCRIPTION('SOFTWARE PRODUCTS') PREFIX(PP*) NAME(GENERAL)
```

Figure 11-6 Sample pools definitions

For more details about the pools, refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405.

11.7.2 Scratch pooling

With OS/390 V2R10, DFSMSrmm extends the choices you have when implementing scratch pooling. You can use any combination of the following and can decide at individual request level which method is used:

- Use DFSMSrmm system-based pooling **VLPOOL** command with SYSID. This enforces pooling based on prefix pools.
- Use DFSMSrmm EDGUX100 installation exit to select a pool prefix. You can use any of the information in the system to make a decision about the pool to be used.
- Use pre-ACS processing to obtain the DFSMSrmm system based pool prefix, or the EDGUX100 exit selected pool prefix in the MSPOOL read-only variable.

Archived

- Use SMS ACS processing to base scratch pooling on tape storage group names. This support is provided for non-system managed tape and for system-managed tapes libraries:
  - For system-managed tape libraries, a SG corresponds with a library.
  - For non-system managed tape, a SG corresponds with a scratch pool.

The order in which DFSMSrmm validates the scratch mount is:

1. If it is a system-managed ATL, leave validation to the system.
2. Storage group assigned for call to SMS ACS routines
3. Pool assigned by EDGUX100 exit
4. DFSMSrmm system based pooling (VLPOOL)

11.8 Delete non-existent volumes

When you have converted to DFSMSrmm, you may have converted information about volumes that you no longer have. Some vendor tape management systems use databases that must be formatted for each range of volumes to be managed. You may have stopped using some of these ranges but never reformatted the database to remove the volumes.

Running under DFSMSrmm, you have the flexibility to add and delete volumes or other resources at any time. Use the knowledge you have about your existing tape volume ranges to check that you have defined to DFSMSrmm only the volumes and rack numbers that you want to manage.

You can use the DFSMSrmm commands to build lists of commands to delete resources you do not want. When you delete volumes, consider deleting the rack numbers that are left in the empty status.

For example, you know that volumes 100000 through 199999 no longer exist but that they were converted to DFSMSrmm. Use the commands in Figure 11-34 to delete the volumes and rack numbers.

Example 11-34 Deleting nonexistent volumes

```
RMM SEARCHVOLUME VOLUME(1*) OWNER(*) LIMIT(*) -
    NOLIST CLIST('RMM DV ',' FORCE')
EXEC EXEC.RMM LIST
RMM DELETERACK 100000 COUNT(99999)
RMM DELETERACK 199999 COUNT(1)
```

Refer to the *DFSMSrmm Implementation and Customization Guide*, SC26-7405, for information about the commands used. You can repeat these commands for each range of volumes and rack numbers you want to delete.

You can avoid the need to do cleanup by deleting the volume information during conversion. Use the EDGCNXIT user exit from EDGCNVT to check for ranges of volumes you want deleted, and tell EDGCNVT to skip the volumes and data set records within those ranges.

11.9 Repairing actions

In a system-managed tape library environment, there are three places to record the tape volume information:

- **Library manager database (LM DB)**
  LM DB records information about all tape volumes in the tape library, including logical volumes in order to control the IBM tape library.
- **Tape configuration database (TCDB)**
  TCDB is a ICF catalog that contains information of tape volumes, which are in the connected tape libraries. TCDB is always required when you control IBM tape libraries with DFSMS regardless of the existence of DFSMSrmm. TCDB is not needed if you plan to use basic tape library support to control your tape libraries.
- **DFSMSrmm control data set**
  DFSMSrmm uses this data set to record all volumes, not only tape library volumes, but also non-tape library tape volumes. More information is recorded in CDS than in TCDB. The control data set also contains information about tape data sets.

All tape volume information in the LM DB, TCDB and DFSMSrmm CDS should be consistent. DFSMSrmm provides the EDGUTIL utility for you to maintain the CDS. Also, you can use the ISMF AUDIT function to compare TCDB with LM DB. Figure 11-7 shows the difference between the audit process before Release 10 and with Release 10.

Archived

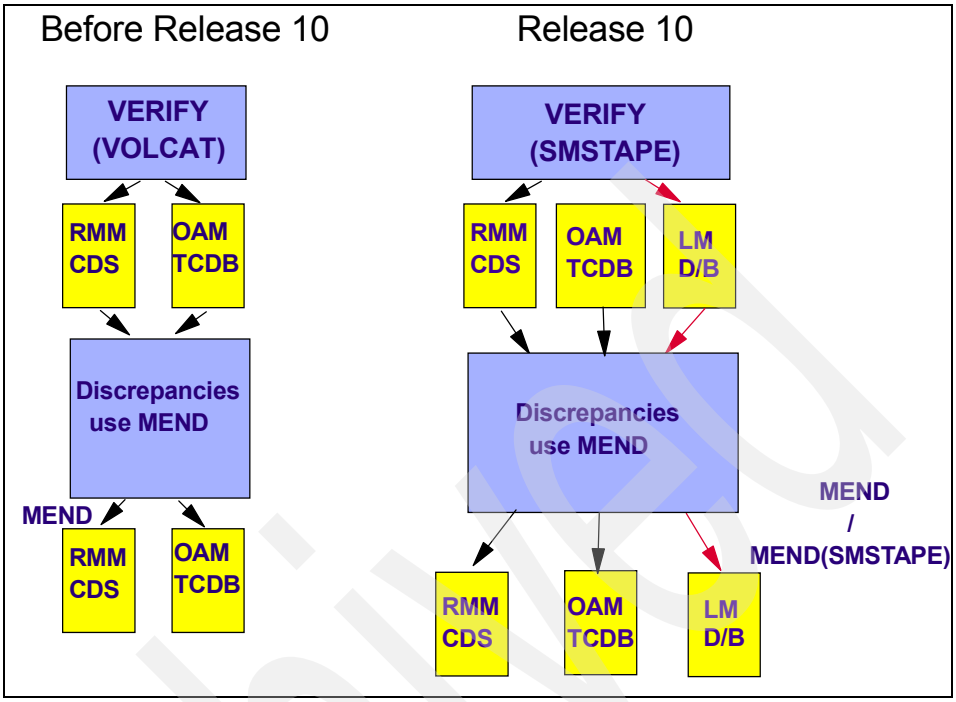


Figure 11-7 Three-way audit of CDS, TCDB and LM DB

11.9.1 Verify

You can use the EDGUTIL utility with the VERIFY parameter to verify either active or inactive CDS. Figure 11-8 shows the VERIFY parameter syntax.

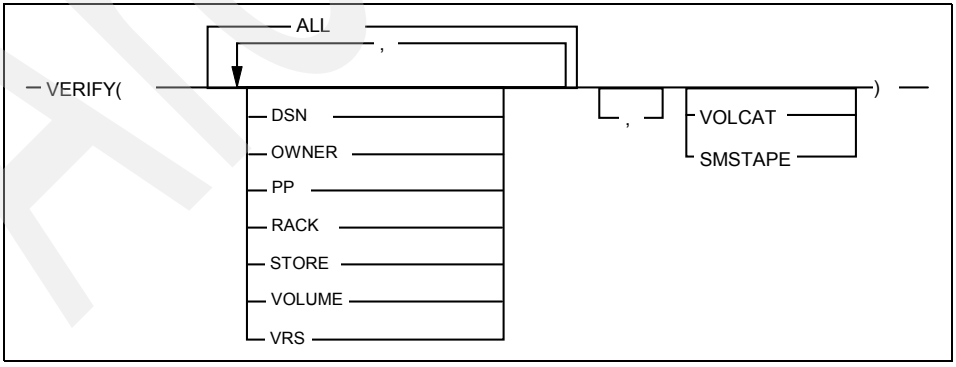


Figure 11-8 EDGUTIL MEND syntax diagram

Verify processing can check the consistency between the records inside the DFSMSrmm CDS. Also:

- When you specify VERIFY(VOLCAT), DFSMSrmm checks the information between CDS and TCDB.
- When you specify VERIFY(SMSTAPE), DFSMSrmm checks the synchronization of the DFSMSrmm CDS with the TCDB, and in addition checks the LM DB if needed.

  For VERIFY(SMSTAPE), DFSMSrmm scans both the DFSMSrmm control data set and the TCDB sequentially to ensure that volume records reside in both of them. If any system-managed volumes are only defined to DFSMSrmm CDS or the TCDB, DFSMSrmm checks it against the LM DB.

  DFSMSrmm also checks stacked volume information against the LM DB. Because the stacked volumes are not system managed volumes, no information about stacked volumes resides in the TCDB. However, DFSMSrmm knows they should be in the LM DB.

  This function is available with the release 10 enhancement, and is meant to be the replacement of VERIFY(VOLCAT).

Example 11-35 shows a sample JCL to verify all information of the CDS.

Example 11-35 JCL to verity all information in CDS

```
//YCJRES4L JOB ......
//UTIL     EXEC PGM=EDGUTIL,PARM='VERIFY'
//SYSPRINT DD DSN=RMM.YCJ4.MESSAGE.UTIL,DISP=SHR
//MASTER   DD DSN=RMM.CONTROL.DSET,DISP=SHR
//SYSIN    DD DUMMY
```

The default value for the VERIFY operand is ALL, but you can verify a specific type of record with CDS, such as the data set information with operand DSN, and so on.

With the enhancement of DFSMSrmm Release 10, you can perform a cross-check against DFSMSrmm, TCDB and LM DB.

11.9.2 Mend

You can only perform the MEND process with an inactive CDS.

When using the MEND parameter, the same verification as with VERIFY is performed and any error status found is corrected. Information in LM DB or TCDB is not checked before the DFSMSrmm Release 10 enhancement.

Archived

If you have successfully enabled stacked volume support with the EDGUTIL UPDATE option STACKEDVOLUME(YES), during MEND processing, DFSMSrmm creates stacked volumes in the DFSMSrmm control data set.

When you run MEND for the first time after enabling stacked volume support, DFSMSrmm needs MEDIANAME information for the stacked volumes it creates from the container information. MEDIANAME information is defined in the VLPOOL statements in PARMLIB. To ensure that this is available for EDGUTIL, you must have started DFSMSrmm at least once with the correct VLPOOL definitions before running EDGUTIL MEND.

Figure 11-9 shows the MEND parameter syntax.

Restriction: Do not enable stacked volume support until all systems using the control data set are on a supporting release level.

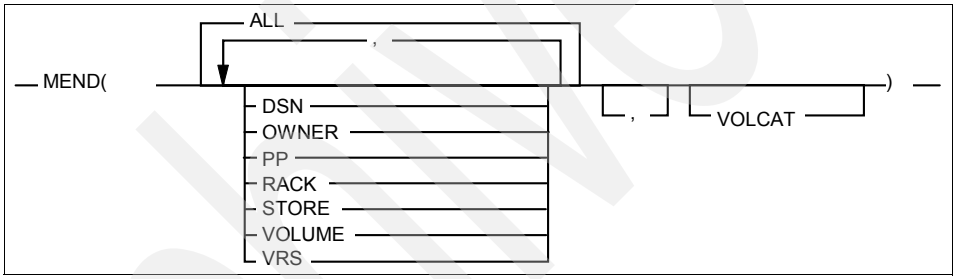


Figure 11-9 EDGUTIL MEND syntax diagram

The default value for MEND operand is ALL, however you can mend a subset of specific record types to increase the performance. Example 11-36 shows sample JCL to MEND data set and volume records.

Example 11-36 JCL to MEND data set and volume records only

```
//YCJRES4L JOB  CLASS=A,MSGCLASS=X,
//         NOTIFY=&SYSUID,TIME=1440
//UTIL     EXEC PGM=EDGUTIL,PARM='MEND(VOLUME,DSN)'
//SYSPRINT DD DSN=RMM.YCJ4.MESSAGE.UTIL,DISP=SHR
//MASTER   DD DSN=RMM.CONTROL.DSET,DISP=SHR
```

After MEND processing is completed, you might need to add or change some volume information. Use the **RMM CHANGEVOLUME** subcommand to set or change any information.

Archived

With the DFSMSrmm Release 10 enhancement, MEND processing can also correct CDS information from TCDB or LM DB. Optionally, it can create a stacked volume record in order to support the VTS enhanced support and perform a 3-way audit to check the information about stacked volumes and logical volumes (see Figure 11-10).

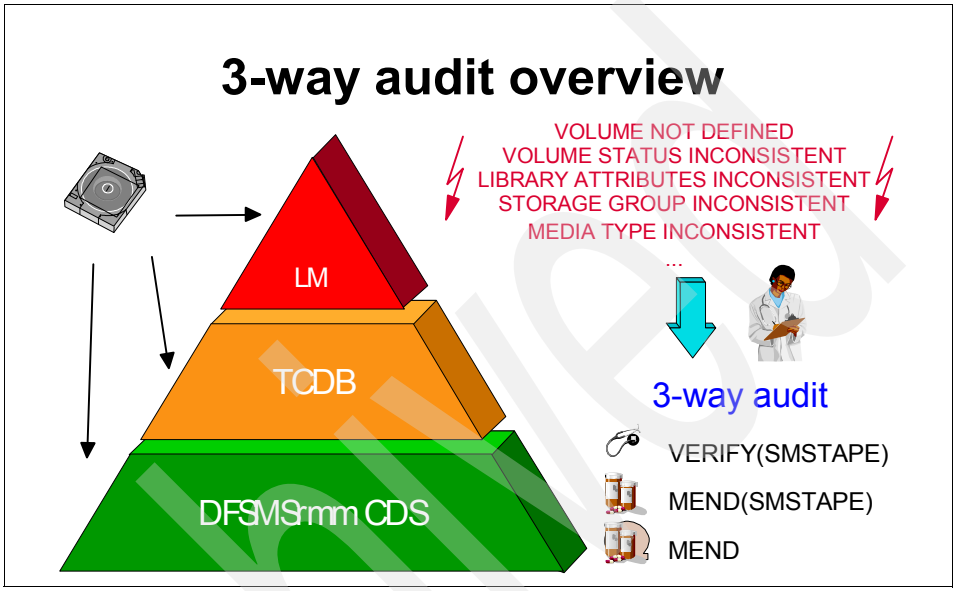


Figure 11-10 DFSMSrmm CDS,Volume Catalog,Library Manager constancy check

When you specify MEND(SMSTAPE), DFSMSrmm performs the same verification action as VERIFY(SMSTAPE) and corrects any error status found. This processing uses information from DFSMSrmm CDS for volume status and storage groups to correct the TCDB and LM DB. Figure 11-11 compares the MEND and MEND(SMSTAPE) processing.

Archive

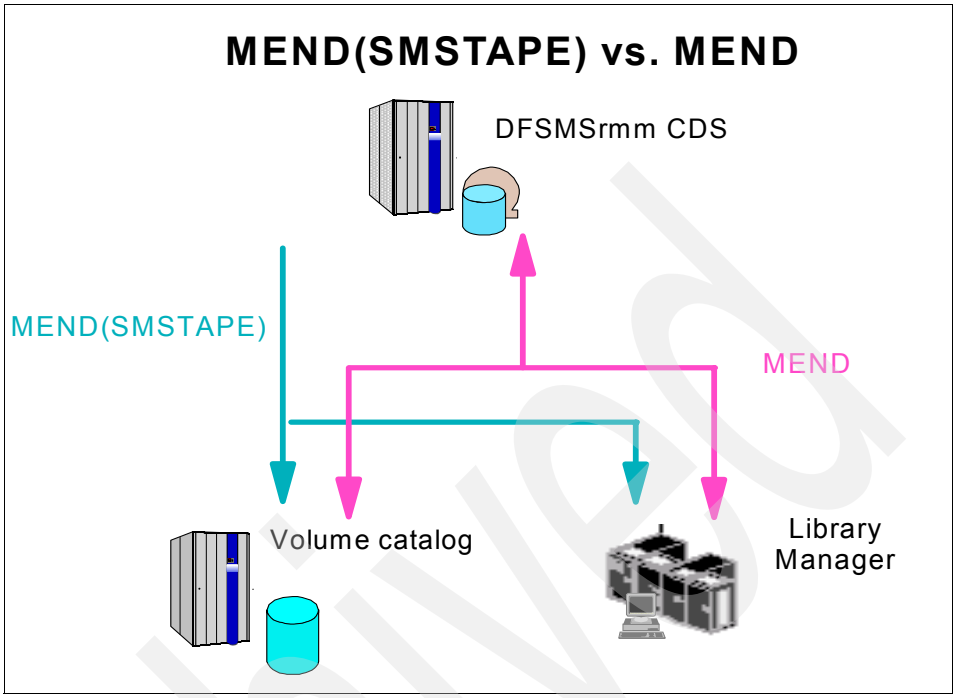


Figure 11-11 EDGUTIL MEND(SMSTAPE) does not update the CDS

You can run MEND(SMSTAPE) processing against either an active or inactive CDS. This function is available with the DFSMSrmm Release 10 enhancement.

11.9.3 DFSMSrmm CDS and DFSMShsm consistency checking

As mentioned at the beginning of this chapter, the DFSMSrmm CDS is the most important part of your DFSMSrmm environment. Ensuring availability of your DFSMSrmm controlled data requires ensuring the health of your control data set. This assurance can be achieved by running the standard DFSMSrmm EDGUTIL VERIFY against the DFSMSrmm CDS. You can also run Mainstar: *Fast*Audit/390 Tape Audit facility, a selectable unit of the Mainstar:*Fast*Audit/390 product suite.

Mainstar's *Fast*Audit/390 product is widely known and used in the z/OS and OS/390 environment. The Tape Audit function was developed to ensure that your DFSMSrmm CDS is healthy, in good shape, and that all data controlled by DFSMSrmm can be accessed when needed. One of the specific audits available and performed is:

- The extracts of all references to tape from DFSMShsm CDSs (except ABARS tape) and compared to the DFSMSrmm CDS.

Other DFSMSrmm diagnostics are:

- Finding tapes in TTOC and MCT/MCD, but not in DFSMSrmm.
- Finding tapes that are permanent in DFSMSrmm, but not in TTOC.
- Finding tapes that are in DFSMSrmm, valid in DFSMShsm, but no entries are valid entries.
- Finding tapes that are in DFSMShsm but expired in DFSMSrmm.
- Finding tapes that are active DFSMSrmm scratch status.

The audits are performed by submission of a single batch job. This batch job performs a multiple of standard equivalent DFSMShsm command audits without adding the additional increase to the system, DFSMSrmm or your DFSMShsm environments. The key to finding the errors is to correctly verify and report these errors accurately. Mainstar's Tape Audit accomplishes this by doing a direct catalog lookup for every instance an error is located, meaning the product is bi-directional on finding, verifying and reporting errors. Once the errors are verified, Mainstar's *Fast*Audit/390 provides an ISPF interface for easy and organized viewing of the errors and files in question. The *Fast*Audit/390 product is shipped with easy to use and understand fixes to assist in correcting the errors. The customer has the ability to create, save and submit custom fixes too. All errors, discrepancies, fixes, and output are easily accessible and viewed through the product ISPF interface panel for future reference.

For more information on *Fast*Audit/390, visit Mainstar's Web site at:

`http://www.mainstar.com`

11.9.4 Using DFSMSrmm with BTLS

You can also use DFSMSrmm to manage volumes in any automated tape library that has special software including an IBM Tape Library dataserver that is managed using Basic Tape Library Support (BTLS).

Both system-managed tape or BTLS support the use of IBM 3494 and 3495 Automated Tape Library Dataservers. DFSMSrmm is integrated with system-managed tape but has no direct interface with BTLS.

You can use DFSMSrmm to work with BTLS to:

- Select volumes from scratch pools
- Return volumes to scratch status

Select scratch pool

BTLS provides a way of selecting from any one of its eight scratch pools using job names. If you already use the data set name to select a scratch pool, you can use DFSMSrmm to select the scratch pool, and DFSMSrmm tells BTLS which pool is to be used. Use the DFSMSrmm **`VLPOOL`** command in PARMLIB to

associate an DFSMSrmm scratch pool with the name of a BTLS scratch pool. For example, we can have the following scratch pool definitions, as shown in Example 11-37.

Example 11-37 Scratch pool definitions

```
VLPOOL PREFIX(10*) TYPE(S) MEDIANAME(CARTS) NAME(SCRTCH1) -
       DESCRIPTION('BTLS pool 1') EXPDTCHECK(N)
VLPOOL PREFIX(2*) TYPE(S) MEDIANAME(3590) NAME(SCRTCH7) -
       DESCRIPTION('BTLS pool 7') EXPDTCHECK(N)
VLPOOL PREFIX(*) TYPE(S) MEDIANAME(TAPE) -
       DESCRIPTION('default pool') EXPDTCHECK(N)
```

To use the data set name to select the scratch pool, you must implement the EDGUX100 installation exit. The *DFSMSrmm Installation and Customization Guide,* SC26-7405, provides details on how to customize, install, and use this exit.

When BTLS issues a mount to the library for a nonspecific volume, DFSMSrmm uses the IGXMSGEX installation exit to return the correct BTLS scratch pool name to BTLS, and BTLS uses that name instead of any pool it may have selected.

Return volumes to scratch status

Although BTLS tracks the status of volumes, changing them to private status when they are used for a nonspecific mount, it provides no method of deciding when to return the volume to scratch. You use your tape management system to do this. DFSMSrmm provides simple TSO commands that you can use to drive the **BTLS LIBRARY** command to change a volume back to scratch status. When you have run DFSMSrmm inventory management, you can generate lists of volumes in scratch status and use them as input to the **BTLS LIBRARY** command to update BTLS volume status. See Example 11-38 for details.

Example 11-38 BTLS status synchronization

```
//SYNCBTLS PROC
//TMP      EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//*
//AMS      EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//LIBIN    DD   DISP=SHR,DSN=userid.EXEC.RMM.CLIST
//         PEND
//S1      EXEC SYNCHBTLS
//TMP.SYSTSIN DD *
RMM SEARCHVOLUME VOLUME(10*) OWNER(*) LIMIT(*) STATUS(SCRATCH) -
NOLIST CLIST
```

```
/*
//AMS.SYSIN DD *
LIBRARY SETCAT UNIT(123) CATEGORY(SCRTCH1)
/*
//S7       EXEC SYNCHBTLS
//TMP.SYSTSIN DD *
RMM SEARCHVOLUME VOLUME(2*) OWNER(*) LIMIT(*) STATUS(SCRATCH) -
NOLIST CLIST
/*
//AMS.SYSIN DD *
LIBRARY SETCAT UNIT(123) CATEGORY(SCRTCH7)
/*
//SM EXEC SYNCHBTLS
//TMP.SYSTSIN DD *
RMM SEARCHVOLUME VOLUME(*) OWNER(*) LIMIT(*) STATUS(MASTER) -
LOCATION(SHELF) INTRANSIT(N) NOLIST CLIST
/*
//AMS.SYSIN DD *
LIBRARY SETCAT UNIT(123) CATEGORY(PRIVATE)
/*
//EJECT EXEC SYNCHBTLS
//TMP.SYSTSIN DD *
RMM SEARCHVOLUME VOLUME(*) OWNER(*) LIMIT(*) DEST(VAULT1) -
LOCATION(SHELF) INTRANSIT(Y) NOLIST CLIST
/*
//AMS.SYSIN DD *
LIBRARY SETCAT UNIT(123) CATEGORY(XEJECTB)
/*
```

By customizing the RMM TSO command, you can select volumes from a single scratch pool if necessary so that BTLS only returns a small number of volumes to scratch status. Similarly, you can use DFSMSrmm commands to build eject lists.

11.9.5 Using DFSMSrmm with non-IBM libraries

Whatever tape library is installed in your system, DFSMSrmm can support it through a user interface that tells the library which volumes to return to scratch status and which volumes must be ejected and moved to another storage location.

With DFSMSrmm's reporting and command capabilities it is simple to build this interface, using REXX, DFSORT, DFSMSrmm utilities, and RMM TSO subcommands.

You must analyze what the vendor tape library requires as input for retention and movement processing and then customize DFSMSrmm to provide the library with the correct input.

Archived

Be sure to keep the vendor tape library catalog and the DFSMSrmm database synchronized to avoid any mismatch that would cause rejection of tape mounts.

STK tape libraries

Below we provide you with a simple interface example between an STK Silo tape library and DFSMSrmm to eject tapes that need to be moved to another location.

After running DFSMSrmm inventory management (storage location management processing DSTORE), you can create a list of all volumes that are moving by using DFSMSrmm commands. Follow these steps:

1. Use the DFSMSrmm commands as shown in Example 11-39:

Example 11-39 Search for all volume pending move

```
RMM SV VOLUME(*) DESTINATION( LOCAL) CLIST('','-')
RMM SV VOLUME(*) DESTINATION( REMOTE) CLIST('','-')
RMM SV VOLUME(*) DESTINATION( DISTANT) CLIST('','-')
RMM SV VOLUME(*) DESTINATION( my_loc) CLIST('','-')
```

> **Note:** Unless you exploit the capability of APAR OW30790 and use a pre-allocated RMMCLIST DD name, you can use only one RMM TSO subcommand at a time, because each call to the CLIST option will overwrite the data created previously.

The resulting list, which is stored in the RMMCLIST data set, or the data set prefix.EXEC.RMM.CLIST, consists of VOLSERs only. The data set uses blocked, variable-length records, with a logical record length of 255. The prefix is the TSO user PROFILE PREFIX that the user has currently defined. If you have specified NOPREFIX, DFSMSrmm allocates a data set using the RACF user ID or the jobname, to avoid using EXEC.RMM.CLIST.

If you are using the RMM TSO subcommands in a batch job, you can set the prefix in front of the RMM TSO subcommand. Figure 169 on page 465 shows sample JCL:

2. Convert the list from variable-length records to fixed-length records. HSC/MVS control statement syntax conventions: the control statement for each utility program consists of a command (indicating the utility function) followed by parameters, as applicable, in 80-character card-image records (see the *HSC/MVS Programmer's Guide*, *Volume 2* provided with the product).
3. Edit this list and create an `EJECT` statement by following Example 11-40.

Example 11-40 Sample EJECT statement example

```
EJECT VOLSER( -
```

```
          volser1 -
          volser2 -
          volser3 -
          ......
     -) CAP( cap-list)
```

Be aware that each single **EJECT** command causes an operator interaction.

4. Run the HSC update utility, SLUADMIN (see the *HSC/MVS System Programmer's Guide, Volume 1* provided with the product). The HSC Initialize Cartridge Utility (SLUADMIN) Figure 11-41 shows sample JCL that you can use to initialize cartridges with the HSC SLUADMIN utility, by using DFSMSrmm EDGINERS processing.

Example 11-41 JCL for initializing cartridges with EDGINERS

```
//YCJRES1  JOB ....
//TINIT    EXEC PGM=SLUADMIN,TIME=1440,PARM=MIXED
//TAPE     DD   UNIT=(580,1,DEFER),DISP=NEW
//* EDGINERS TAPE UNIT DD
//SLSTAPE  DD  UNIT=AFF=TAPE INIT CARTRIDGES INIT TAPE
//* UNIT DD.
//SYSIN    DD   UNIT=VIO,SPACE=(TRK,1) EDGINERS CONTROL DATASET
//SLSPRINT DD   SYSOUT=* UTILITY MESSAGES
//SYSPRINT DD   SYSOUT=* EDGINERS MESSAGES
//SLSINIT  DD   *
   INIT LABEL(SL) VOL(******)
//*
//SLSIN    DD   *
* WHEN USING THE DFSMSRMM TAPE INITIALIZATION PROGRAM EDGINERS, YOU
* MUST SPECIFY THE 'PROGRAM' PARAMETER ON THE INIT CARTRIDGES
* CONTROL STATEMENT.
 INITIALIZE CAP(000) PROGRAM(EDGINERS) OPTION(SCRATCH) CNTLDD(SYSIN)
/*
```

The HSC scratch conversion SLUCONDB program is provided to generate scratch transactions from DFSMSrmm tape management system. SLUCONDB calls the routine SLUDRRMM to read the DFSMSrmm report extract file if RMM is specified in the EXEC JCL statement. This program utilizes the DFSMSrmm EDGRVEXT macro to map the layout of the extract file. The following list shows the fields that are used:

- RVTYPE
- RVVOLSER
- RVLCDATE
- RVNAME
- RVSTATUS
- RVEXPDT

- RVEXPDTO
- RVLABEL

SLUDRRMM processes the extract and passes the volume record information to SLUCONDB, which builds scratch card images to input to SLUADMIN. Volume status scratch (scratch or non scratch) is then updated in the HSC CDS for each volume record in the extract file. Dates on the DFSMSrmm report must be in Julian date format (EDGHSKP run with PARM ′ DATEFORM(J)′). Tapes listed on the DFSMSrmm report without expiration dates will be skipped by SLUDRRMM. Example 11-42 is an example of the DFSMSrmm and HSC synchronization JCL.

Example 11-42 DFSMSrmm and HSC synchronization JCL

```
//JOBSCUP JOB (ACCOUNT),PROGRAMMER
//S1 EXEC PGM=EDGHSKP,PARM=¢ RPTEXT,DATEFORM(J)¢
//MESSAGE DD DSN=RMM.MESSAGE.DATASET,DISP=SHR
//REPTEXT DD DSN=RMM.REPORT.EXTRACT,DISP=SHR
//S2 EXEC PGM=SLUCONDB,PARM=('RMM,SCRPOOL(SL)',
'87100-87110,MIXED')
//SLSTMS DD DSN=RMM.REPORT.EXTRACT,DISP=SHR
//SLSSOUT DD DSN=&&SCUPINPT,DISP=(NEW,PASS),UNIT=,SPACE=DCB=
//SLSPRINT DD SYSOUT=A
//S3 EXEC PGM=SLUADMIN
//SLSIN DD DSN=&&SCUPINPT,DISP=(OLD,DELETE)
//SLSPRINT DD SYSOUT=A
//
```

The SLUCONDB utility, when PARM=RMM is specified, checks in the DFSMSrmm extract file to see if the MEDIANAME of the tapes being returned to scratch is either 3480 or 3490. For example, if you changed the default media name 3480, and you are using CARTS, you will never get your tapes selected as scratch by the utility.

STK provides a sample insert/delete user exit SLSUX06, this exit enables the HSC to get control when the database is being updated to add or delete a volume. Refer to STK tape libraries documentation for details on how to tailor this exit to work with IBM DFSMSrmm.

Archived

12

Reporting on your DFSMSrmm environment

In this chapter we describe how you can use different ways to create DFSMSrmm reports or get DFSMSrmm information.

Objectives of this chapter:

- Learn to use the RMM TSO subcommands or the DFSMSrmm ISPF dialog to retrieve information on your DFSMSrmm environment.
- Use the DFSMSrmm-supplied standard reports for reporting requirements to view online or printed reports on a regular basis.
- Set up and use the DFSMSrmm report generator.
- Produce reports using a sort utility like DFSORT or DFSORT's ICETOOL to create customized reports from the available DFSMSrmm information, such as the extract data set, the activity file, and System Management Facility (SMF) records.

In the following sections, we provide detailed information about all the different ways to get information from DFSMSrmm. We include (at the end of the chapter) sample JCL for generating some useful reports.

© Copyright IBM Corp. 2004. All rights reserved.

12.1 Requesting DFSMSrmm resource information

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 information.

You can use the DFSMSrmm ISPF dialog or the RMM TSO subcommands to obtain information about your resources. You can also use the ISPF fast path commands to reach the correct ISPF panels for requesting the desired information. For more information about all the available fast path commands, refer to *z/OS DFSMSrmm Guide and Reference*, SC26-7404.

The RMM TSO subcommands and the DFSMSrmm ISPF dialog share some similarities. The dialog allows you to view the information in real time and in predefined formats. The dialog also allows you to decide dynamically which further details you want to view. You can use the RMM TSO subcommands to obtain the kind of information that you obtain when you use the DFSMSrmm ISPF dialog. The difference is that your do not have full-screen viewing capability when you use the RMM TSO subcommands. You can use the commands interactively or submit them in batch. You can save the batch job input, which allows you to reuse the commands, so you can run the job when it is required.

You can get information about your resources in two different formats:

- List format. You can generate lists of resources based on a selection criteria.
- Display format. This format provides to you the detailed information about a specific resource.

In the following sections, you will learn how to use the different interfaces for getting the information that you need.

12.1.1 Working with resource lists

Any user can request a list of resources defined to DFSMSrmm. You use the DFSMSrmm ISPF panels or RMM TSO subcommands to define criteria for the list. You can change the sort order of some lists either before or after DFSMSrmm builds them.

Depending on your level of authorization, you can enter line operators against items in some lists to request additional information, or request certain functions.

You can request lists of shelf locations, volumes, software products, data sets, and vital record specifications.

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, and request detailed information about it through the Display panels.

Creating lists

You can create lists of resources using ISPF Search panels or the RMM TSO **SEARCH** subcommands.

You can issue the RMM TSO subcommands from a batch job, using the JCL shown in Example 12-1.

Example 12-1 ample JCL to issue RMM TSO subcommands

```
//YCJRES4X JOB MSGCLASS=X,NOTIFY=&SYSUID
//*  INIT VALIDATE
//TMPBATCH EXEC PGM=IKJEFT01,REGION=4M,DYNAMNBR=10
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
RMM subcommand
/*
```

You can also reach the search panels using the following fast path commands from the command line:

- **RACK SEARCH**
- **VOLUME SEARCH**
- **PRODUCT SEARCH**
- **DATASET SEARCH**
- **VRS SEARCH**

Shelf location lists

You can create shelf location lists from ISPF panels or through the RMM TSO **SEARCHRACK** subcommand.

You can reach the DFSMSrmm Rack and Bins Search panel following any of the paths shown in Figure 12-1.

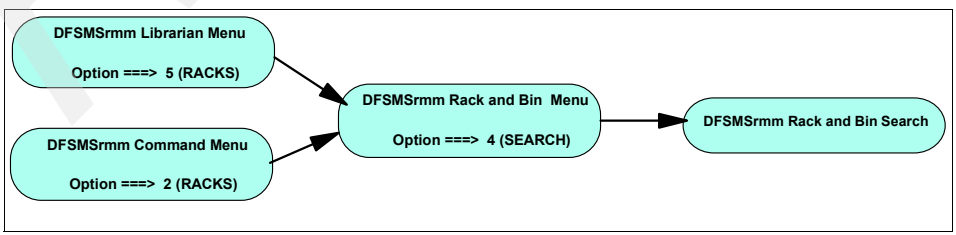


Figure 12-1 Reaching the Rack and Bin Search panel

Archived

Once you are in the DFSMSrmm Rack and Bin Search panel, you must enter one of the selections of the following information for the search criteria:

- Rack or bin number
- Location
- Status
- Media name or pool ID

In Figure 12-2, you can see an example of searching for all the shelf locations in SCRATCH status.

```
 Panel  Help
------------------------------------------------------------------------------
                          DFSMSrmm Rack and Bin Search
Command ===>

Specify one of:
  Rack number . . . . . *          May be generic.
    or
  Bin number  . . . . .            May be generic.

Location  . . . . . .              SHELF, SMS library name, or storage
                                   location
Media name  . . . . . .            Limit to a single media name

Pool  . . . . . .                  or to a particular pool

Status  . . . . . . . SCRATCH      EMPTY, IN USE or SCRATCH

Limit . . . . . . . . 5            Limit to first nnnn racks or bins.
                                 Default = 10.
```

Figure 12-2 Searching for shelf locations

In Figure 12-3, you can see the generated list with the shelf locations that meets the selection criteria you entered in the search panel. In this case, it is a list of the shelf locations in SCRATCH status. Because we did not enter any value in the Limit field, DFSMSrmm only shows the first ten shelf locations.

Archived

```
 Panel  Help
------------------------------------------------------------------------------
                            DFSMSrmm Racks or Bins           Row 1 to 5  of  5
Command ===>                                                  Scroll ===> CSR

The following line command is valid: V.
  Rack or Bin Media    Volume
S Number      Name     Serial  Status
- ----------- -------- ------  -------
  PP0001      CARTS    DZ11C0  SCRATCH
  PP0002      CARTS    DZ11C1  SCRATCH
  HGT001      3490     HGT001  SCRATCH
  HGT002      3490     HGT002  SCRATCH
  HGT003      3490     HGT003  SCRATCH
****************************** Bottom of data ********************************
```

Figure 12-3 List of SCRATCH shelf locations

To display a list of shelf locations, you can also use the RMM TSO **SEARCHRACK** command from a command line or from a JCL.

Example 12-2 shows you an example command for generating the same list as in the previous ISPF example.

Example 12-2 TSO search rack

```
RMM SEARCHRACK RACK(*) SCRATCH
```

The output from this command is shown in Figure 12-4.

```
Rack    Medianame  Volume  Status    Location
------  ---------  ------ --------  --------
PP0001  CARTS      DZ11C0  SCRATCH   SHELF
PP0002  CARTS      DZ11C1  SCRATCH   SHELF
HGT001  3490       HGT001  SCRATCH   SHELF
HGT002  3490       HGT002  SCRATCH   SHELF
HGT003  3490       HGT003  SCRATCH   SHELF
SEARCH COMPLETE - MORE ENTRIES MAY EXIST
 5         ENTRIES LISTED
```

Figure 12-4 RMM TSO SEARCHRACK command output

Volume lists

You can reach the DFSMSrmm Volume Search panel by following any of the paths shown in Figure 12-5.

Archived

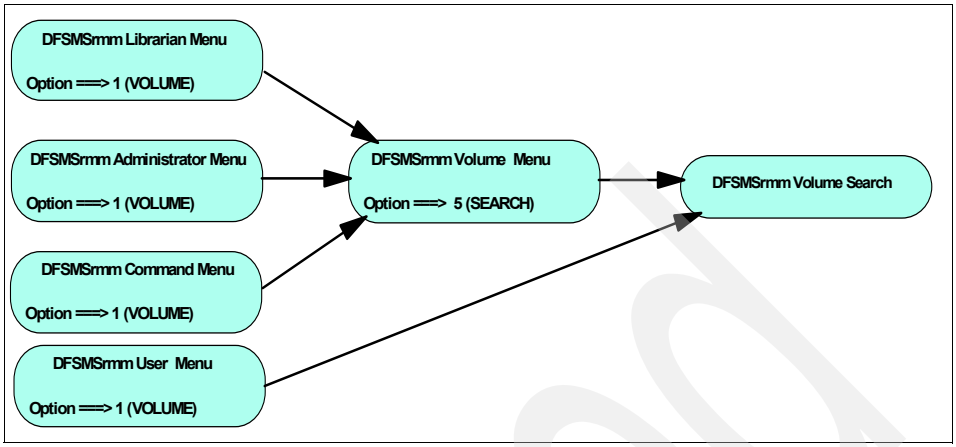


Figure 12-5 Reaching the Volume Search panel

Once you are in the DFSMSrmm Volume Search panel, you must enter one of the following pieces of information as the search criteria selection:

- Volume serial number
- Volume owner
- Media name or pool ID
- Status
- Date assigned
- Volume retention date
- Volume home location
- Volume location
- Stacked volume container
- Volume type
- Media type
- Volume label type
- Current ANSI label version
- Required ANSI label version
- Tape density
- Tape format
- Compaction
- Volume attributes
- Volume destination
- Required location
- Move mode
- Intransit

In Figure 12-6, you can see the Volume Search panel. We have selected all the volumes owned by YCJRES1, and we limit the number of list entries to 15.

Archived

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

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

Owner . . . . . YCJRES1     Owned by a specific user. Default is your userid

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

Status  . . . .
                            Select volume status. Default is ALL.
Since . . . . .             Volumes assigned since YYYY/DDD
Retention . . .             Volumes retained up to YYYY/DDD
Limit . . . . . 5           Limit search to first nnnn volumes

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

Volume type . .             ( LOGICAL , PHYSICAL or STACKED )
Media type  . .             Tape media type ( *, CST, ECCST, HPCT or EHPCT )
Label . . . . .             Tape label standard ( for example SL )
  Current version           Label version number( for example 3 )
  Required version          Label version number( for example 4 )
Density . . . .             Tape recording density
Format  . . . .             Tape recording format ( *, 18TRACK, 36TRACK,
                                                    128TRACK or 256TRACK )

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

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

The following line commands will be available when the list is displayed:
  C - Change volume information            E - Eject volume
  I - Data set information                  L - List volume chain
```

Figure 12-6 Searching for volumes

In Figure 12-7, you can see the first page of the generated volumes list. By pressing the PF11 (RIGHT) key, you can go to the second page for more information columns, as shown in Figure 12-8.

Archived

```
 Panel  Help  Scroll
------------------------------------------------------------------------------
                  DFSMSrmm Volumes (Page 1 of 2)           Row 1 to 15 of 15
Command ===>                                                Scroll ===> CSR

The following line commands are valid: C,I,O,R,E,L and V
Use the RIGHT command to view other data columns
  Volume          Assigned   Expiration                  Dest-    Tra- Data
S serial Owner    date       date       Status   Location ination  nsit sets
- ------ -------- ---------- ---------- -------- -------- -------- ---- ----
  HGT000 YCJRES1  2001/060   2001/060   USER     SHELF             N    1
  TST010 YCJRES1  2001/059   2001/059   MASTER   LIB1              N    1
  TST012 YCJRES1  2001/058   2001/059   MASTER   LIB1              N    1
  TST013 YCJRES1  2001/058   2001/058   MASTER   LIB1              N    1
  TST015 YCJRES1  2001/059   2001/059   MASTER   LIB1              N    1
   ****************************** Bottom of data ********************************
```

Figure 12-7 Volumes list (1 of 2)

```
 Panel  Help  Scroll
------------------------------------------------------------------------------
                  DFSMSrmm Volumes (Page 2 of 2)           Row 1 to 15 of 15
Command ===>                                                Scroll ===> CSR

The following line commands are valid: C,I,O,R,E,L and V
Use the LEFT command to view other data columns
   Volume Rack   Media    Home     Media          Recorded Compac-  Attri-
S  serial number name     location type     Label format   tion     butes
-- ------ ------ -------- -------- -------- ----- -------- -------- --------
   HGT000 HGT000 3490     SHELF    ECCST    SL    36TRACK  YES      NONE
   TST010 TST010 3590     LIB1     HPCT     SL    128TRACK YES      NONE
   TST012 TST012 3590     LIB1     HPCT     SL    128TRACK YES      NONE
   TST013 TST013 3590     LIB1     HPCT     SL    128TRACK YES      NONE
   TST015 TST015 3590     LIB1     HPCT     SL    128TRACK YES      NONE
```

Figure 12-8 Volumes list (2 of 2)

You can also use the RMM TSO **SEARCHVOLUME** command to display the same volume list. The command in our case should be as follows:

```
RMM SEARCHVOLUME VOLUME(*) OWNER(YCJRES1) LIMIT(5)
```

The output from this command is shown in Figure 12-9.

Archive

```
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- -------
HGT000 YCJRES1  HGT000 2001/060   2001/060   SHELF    1     UV
TST010 YCJRES1  TST010 2001/059   2001/059   LIB1     1     MV
TST012 YCJRES1  TST012 2001/058   2001/059   LIB1     1     MV
TST013 YCJRES1  TST013 2001/058   2001/058   LIB1     1     MV
TST015 YCJRES1  TST015 2001/059   2001/059   LIB1     1     MV       DISTANT
5         ENTRIES LISTED
```

Figure 12-9 RMM SEARCHVOLUME command output

The information returned as output when using RMM subcommands is not as complete as when using the ISPF dialog.

Program products lists

You can reach the DFSMSrmm Product Search panel following any of the paths shown in Figure 12-10.

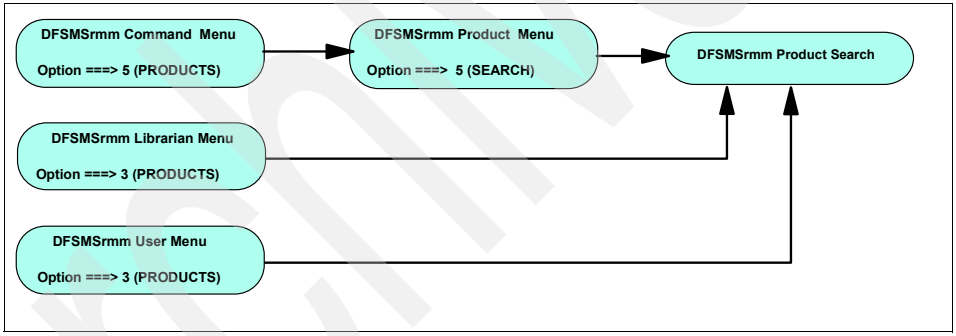


Figure 12-10 Reaching the Product Search panel

Once you are in the DFSMSrmm Product Search panel, you must enter the following information as the search criteria:

- Software product number or name (or generic)

In Figure 12-11, you can see the Product Search panel. We have decided to search all products defined to DFSMSrmm.

Archived

```
 Panel  Help
------------------------------------------------------------------------------
                            DFSMSrmm Product Search
Command ===>

Enter full or partial product number or name.
  Product number . . *

  Product name . . . *

List only the first nnnn products:

  Limit  . . . . . .                Default = 10



The following line commands will be available:
A  - Add a product volume                    P - Display product details
C  - Change product                          R - Release product volume
D  - Delete product                          S - Complete volume details
DN - Delete product without releasing volumes V - Display volume details
```

Figure 12-11 Product Search panel

You can see the generated product list in Figure 12-12 and Figure 12-13.

```
 Panel  Help  Scroll
------------------------------------------------------------------------------
                      DFSMSrmm Products (Page 1 of 2)        Row 1 to 2 of 2
Command ===>                                                 Scroll ===> CSR

The following line commands are valid: A, C, D, DN, P, R, S, V.
Use the RIGHT command to view volume description
S  Product   Level      Product Name                    Vols  Volume  Feature
   Number                                                    Serial  Code
-- --------  ---------  ------------------------------  ----  ------  -------
   PROD01    V01R00M00  TEST FACILITY                   0
   PROD02    V01R01M00  NEW TEST FACILITY               0
****************************** Bottom of data ********************************
```

Figure 12-12 Product list (1 of 2)

Archived

```
 Panel  Help  Scroll
------------------------------------------------------------------------------
                        DFSMSrmm Products (Page 2 of 2)        Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR

The following line commands are valid: A, C, D, DN, P, R, S, V.
Use the LEFT command to view product name
S  Product   Level      Volume description              Vols  Volume  Feature
   Number                                                     Serial  Code
-- --------  ---------  ------------------------------  ----  ------  -------
   PROD01    V01R00M00                                  0
   PROD02    V01R01M00                                  0
****************************** Bottom of data ********************************
```

Figure 12-13 Product list (2 of 2)

You can also request the same product list using the command shown in Example 12-3.

Example 12-3 TSO search product

```
RMM SEARCHPRODUCT
```

The output from this command is shown in Figure 12-14.

```
Number   Level     Product Name                  Feature Code Vols    Volume
-------- --------- ----------------------------- ------------ ----    ------
PROD01   V01R00M00 Test Facility                              0
PROD02   V01R01M00 New Test Facility                          0
2          ENTRIES LISTED
```

Figure 12-14 RMM SEARCHPRODUCT command output

Data set lists

You can reach the DFSMSrmm Data Set Search panel following any of the paths shown in Figure 12-15.

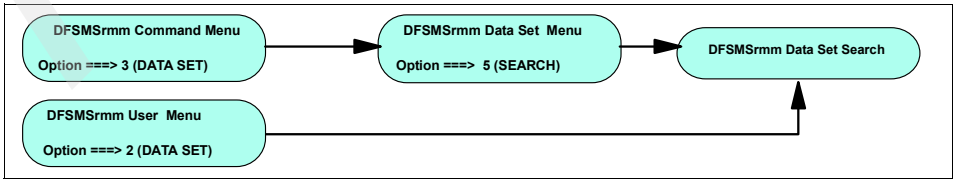


Figure 12-15 Reaching the Data Set Search panel

Archived

Once you are in the DFSMSrmm Data Set Search panel, you must enter one of the following information selections as the search criteria:

- Data set name
- Creating jobname
- Volume owner or volume serial number
- Status
- Retained by vital record specification
- Data set create date
- Data set sequence number
- Creating program name

In Figure 12-16, you can see the Data Set Search panel. We chose to find all the data sets owned by YCJRES1 and retained by VRS. Also, we set the Limit to 20.

```
 Panel  Help
------------------------------------------------------------------------------
                          DFSMSrmm Data Set Search
Command ===>

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

Data set name . . . .
Job name  . . . . . .              Specific or generic name

Enter optional parameters to qualify search
                                                                    More:     +
Owner . . . . . . . . YCJRES1      Owner of volumes (Default is your userid)
Volume serial . . . .            OR Volume serial
Status  . . . . . . .               PRIVATE, SCRATCH, or blank for all
Retained by VRS . . . YES           YES, NO, or blank for all
Create date . . . . .               Created since YYYY/DDD
Physical file seq . .
Limit . . . . . . . . 5           Limit search to first nnnn data sets
Program name  . . . .               Specific or generic name
Create date . . . . .               Created since YYYY/DDD
Physical file seq . .
Limit . . . . . . . . 20            Limit search to first nnnn data sets
Program name  . . . .               Specific or generic name

The following line commands will be available when the list is displayed
  I - Data set information               V - Volume information
  O - Owner information                  C - Change data set information
  D - Delete data set information        L - List multi-volume data set chain
```

Figure 12-16 Data Set Search panel

The generated data set list is shown in Figure 12-17.

Archived

```
 Panel  Help
------------------------------------------------------------------------------
                              DFSMSrmm Data Sets             Row 1 to 5 of 5
Command ===>                                                  Scroll ===> CSR


The following line commands are valid: C, D, I, L, O, V.
                                             Volume          File Create
S Data set name                              Serial Owner    Seq  Date
- ------------------------------------------- ------ -------- ---- ----------
  YCJRES1.DATASET.PRUACS                     HGT000 YCJRES1  1    2001/060
  YCJRES1.DATASET.TEST1                      TST020 YCJRES1  1    2001/058
  YCJRES1.DATASET.VRSNEW.TEST1               TST012 YCJRES1  1    2001/058
  YCJRES1.TEST1.VRS2                         TST016 YCJRES1  1    2001/059
  YCJRES1.TEST1.VRS2                         TST023 YCJRES1  1    2001/060
  ******************************** Bottom of data ********************************
```

Figure 12-17 Data set list

The equivalent command for generating a list of data sets owned by YCJRES1 and retained by VRS is as shown in Example 12-4.

Example 12-4 TSO search data set

```
RMM SEARCHDATASET OWNER(YCJRES1) VITAL(YES) LIMIT(5)
```

The output for this command is shown in Figure 12-18.

```
Data set name                              Volume Owner    Create date Seq
------------------------------------------ ------ -------- ----------- ---
YCJRES1.DATASET.PRUACS                     HGT000 YCJRES1  2001/060    1
YCJRES1.DATASET.TEST1                      TST020 YCJRES1  2001/058    1
YCJRES1.DATASET.VRSNEW.TEST1               TST012 YCJRES1  2001/058    1
YCJRES1.TEST1.VRS2                         TST016 YCJRES1  2001/059    1
YCJRES1.TEST1.VRS2                         TST023 YCJRES1  2001/060    1
5         ENTRIES LISTED
```

Figure 12-18 RMM SEARCHDATASET command output

Virtual record specifications lists

You can reach the DFSMSrmm Data Set Search panel by following any of the paths shown in Figure 12-19.

Archive

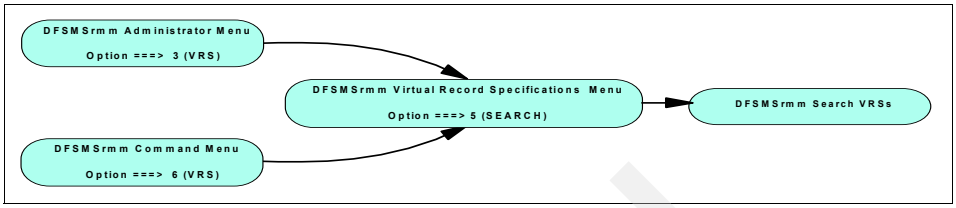


Figure 12-19 Reaching the Search VRSs panel

Once you are in the DFSMSrmm Search VRSs panel, you must enter one of the following information selections as the search criteria:

- Data set name mask
- Jobname mask
- Volume serial number VRS
- Name VRS
- Location
- Next VRS in chain
- VRS owner
- Retention type
- While cataloged
- Until expired
- Expiry date ignore
- Scratch immediate

In Figure 12-20, you can see the Search VRSs panel. We choose to find all the VRSs with Retention type `CYCLES` and While cataloged as `YES`.

```
                          DFSMSrmm Search VRSs
Command ===>

Optionally specify one of:
Data set mask   '**.**'                                    GDG  . .
    Job name mask                                          ( Yes or No )
Volume serial                 Retention type . . . CYCLES
VRS name  . . .               While cataloged  . . YES  ( Yes or No )
                              Until expired  . . .     ( Yes or No )

Location  . . . .             Release options:
Next VRS in chain             Expiry date ignore       ( Yes or No )
  Chain using . .             Scratch immediate        ( Yes or No )

Owner . . . . . *
Limit . . . . . .             ( Limit search to first nnnn VRSs.
                              Default = 10 )
```

Figure 12-20 Search VRSs panel

Archived

The generated VRS list is shown in Figure 12-21, Figure 12-22, and Figure 12-23.

```
                        DFSMSrmm VRSs (Page 1 of 3)          Row 1 to 4 of 4
Command ===>                                                Scroll ===> CSR

The following line commands are valid: C, D, L, M, O, S.
Use the RIGHT command to view other data columns.

S Volume/Data set/Name specification          Job name Type Location Priority
- ------------------------------------------- -------- ---- -------- --------
  *                                                    DSN  HOME     0
  D99000                                               DSN  HOME     0
  YCJRES1.CAT*.**                                      DSN  HOME     0
  YCJRES4.TEST.**                                      DSN  HOME     0
****************************** Bottom of data ******************************
```

Figure 12-21 VRS list (1 of 3)

```
                        DFSMSrmm VRSs (Page 2 of 3)          Row 1 to 4 of 4
Command ===>                                                Scroll ===> CSR

The following line commands are valid: C, D, L, M, O, S.
Use the LEFT and RIGHT commands to view other data columns.
                                              Reten-  Cata- Until Delete
S Volume/Data set/Name specification          tion    loged Expd  Date
- ------------------------------------------- ------- ----- ----- ----------
  *                                           CYCLES  YES   NO    1999/365
  D99000                                      CYCLES  YES    NO     1999/365
  YCJRES1.CAT*.**                             CYCLES  YES   NO     1999/365
  YCJRES4.TEST.**                             CYCLES  YES   NO     1999/365
****************************** Bottom of data ******************************
```

Figure 12-22 VRS list (2 of 3)

Archived

```
                        DFSMSrmm VRSs (Page 3 of 3)          Row 1 to 4 of 4
Command ===>                                                Scroll ===> CSR

The following line commands are valid: C, D, L, M, O, S.
Use the LEFT command to view other data columns.
                                                 Next VRS chain
S Volume/Data set/Name specification             in chain USing Owner    XI  SI
- ------------------------------------------- -------- ----- -------- --- ---
  *                                                                KOHJI    NO  NO
  D99000                                                           YCJRES1  NO  NO
  YCJRES1.CAT*.**                                                  YCJRES1  NO  NO
  YCJRES4.TEST.**                                                  YCJRES4  NO  NO
```

Figure 12-23 VRS list (3 of 3)

If you choose to use the RMM TSO subcommands, then the command you should use is as shown in Example 12-5.

Example 12-5 TSO search vital record specification

```
RMM SEARCHVRS DSNAME('**.**') CYCLES WHILECATALOG OWNER(*)
```

The output from this command is shown in Figure 12-24.

```
Vital Record Specification                Job name Type Location Next VRS
--------------------------                -------- ---- -------- --------
*                                                  DSN  HOME
D99000                                             DSN  HOME
YCJRES1.CAT*.**                                    DSN  HOME
YCJRES4.TEST.**                                    DSN  HOME
4         ENTRIES LISTED
```

Figure 12-24 SEARCHVRS command output

Sorting lists

You can select the sort options for your resource lists before or after you generate them.

If you want to change the sort options for a specific resource before generating the list, then you need to go to the DFSMSrmm Dialog Sort Options Menu. You can reach this menu by selecting **Option 0 (OPTIONS)** in the Removable Media Manager main menu and then **Option 2 (SORT),** or issuing the fast path command **`OPTIONS SORT`** in the command line.

You will reach the Dialog Sort Options Menu. In this panel, you can select the resource you want to change.

In Figure 12-25 and Figure 12-26, you can see an example of how to change the sort options for a volume list.

```
                        DFSMSrmm Dialog Sort Options Menu
Option ===> 4

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.

5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-25 Dialog Sort Options Menu panel

```
                        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     01       A        Media type          02       a
   Status              03       A        Label
   Location            04       A        Recorded format
   Destination                           Compaction
   Transit                               Attributes
   Data sets                             Action
```

Figure 12-26 Volume List Sort Options panel

We have elected to order the volumes based on their expiration date, media type, status, and location. In Figure 12-27, you can see the list generated before changing the sort options, with the default option (sort by volume serial). We have chosen to search all TST*, and we have limited the number of list entries to 10 (the default value).

Archived

```
                    DFSMSrmm Volumes (Page 1 of 2)            Row 1 to 10 of 10
Command ===>                                                   Scroll ===> CSR

The following line commands are valid: C,I,O,R,E,L and V
Use the RIGHT command to view other data columns
  Volume          Assigned   Expiration                  Dest-    Tra- Data
S serial Owner    date       date       Status   Location ination  nsit sets
- ------ -------- ---------- ---------- -------- -------- -------- ---- ----
  TST000 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST001 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST002          2001/064              SCRATCH  LIB1              N    1
  TST003 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST004 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST005          2001/064              SCRATCH  LIB1              N    0
  TST006          2001/065              SCRATCH  LIB1              N    1
  TST007          2001/065              SCRATCH  LIB1              N    1
  TST008 STC      2001/065   2001/065   MASTER   LIB1              N    1
  TST009 STC      2001/065   2001/065   MASTER   LIB1              N    1
****************************** Bottom of data ********************************
```

Figure 12-27 Volume list with default sorting options

In Figure 12-28, you can see the same list after changing the sort options.

```
                    DFSMSrmm Volumes (Page 1 of 2)            Row 1 to 10 of 10
Command ===>                                                   Scroll ===> CSR

The following line commands are valid: C,I,O,R,E,L and V
Use the RIGHT command to view other data columns
  Volume          Assigned   Expiration                  Dest-    Tra- Data
S serial Owner    date       date       Status   Location ination  nsit sets
- ------ -------- ---------- ---------- -------- -------- -------- ---- ----
  TST002          2001/064              SCRATCH  LIB1              N    1
  TST005          2001/064              SCRATCH  LIB1              N    0
  TST006          2001/065              SCRATCH  LIB1              N    1
  TST007          2001/065              SCRATCH  LIB1              N    1
  TST000 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST001 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST003 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST004 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT  N    1
  TST009 STC      2001/065   2001/065   MASTER   LIB1              N    1
  TST008 STC      2001/065   2001/065   MASTER   LIB1              N    1
****************************** Bottom of data ********************************
```

Figure 12-28 Volume list with changed sorting options

Once you have your generated list, you can change the sort option using the **SORT** commands from the command line. Basically, the syntax of the **SORT** command is:

```
SORT <direction> <field>
```

Where `direction` means whether you want to show the values in *ascending* or *descending* order, and `field` is the name of the column you want to sort by.

For example, using the list generated in Figure 12-28, we can choose to sort by the Assigned Date in descending order. In that case, the command should be as follows:

```
SORT DESCENDING ASSIGNED
```

The resulting list is shown in Figure 12-29.

```
                  DFSMSrmm Volumes (Page 1 of 2)           Row 1 to 10 of 10
Command ===>                                                Scroll ===> CSR

The following line commands are valid: C,I,O,R,E,L and V
Use the RIGHT command to view other data columns
  Volume          Assigned   Expiration                      Dest-    Tra- Data
S serial Owner    date       date       Status   Location   ination  nsit sets
- ------ -------- ---------- ---------- -------- -------- -------- ---- ----
  TST009 STC      2001/065   2001/065   MASTER   LIB1                N    1
  TST007          2001/065              SCRATCH  LIB1                N    1
  TST008 STC      2001/065   2001/065   MASTER   LIB1                N    1
  TST006          2001/065              SCRATCH  LIB1                N    1
  TST002          2001/064              SCRATCH  LIB1                N    1
  TST005          2001/064              SCRATCH  LIB1                N    0
  TST004 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT    N    1
  TST003 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT    N    1
  TST001 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT    N    1
  TST000 YCJRES1  2001/061   2001/061   MASTER   LIB1     DISTANT    N    1
****************************** Bottom of data ********************************
```

Figure 12-29 Volume list using SORT command

You can select all the different sorting criteria you want. After that, if you want to show again the original list, you can enter the **SORT** command without any parameters.

For more information about the SORT options, refer to *z/OS DFSMSrmm Guide and Reference*, SC26-7404.

Archive

Using line operators on generated lists

When you generate resource lists, some of them allow you to enter some line operators. Using these line operators, you can obtain more details about one specific resource, or perform a specific action against one resource.

Table 12-1 shows the line operators allowed in each of the resource lists you can generate, and the description of the action each operator performs.

Table 12-1 Resource list line operators

Resource list	Line operator	Description
Shelf location	V	Display volume details
Volume	C	Change volume information
	E	Eject a volume (requires confirmation)
	I	Display volume contents
	L	List multi-volume chain
	O	Display owner information
	R	Release a volume (requires confirmation)
	V	Display volume details
Software products	A	Add product volumes
	C	Change product details
	D	Delete a product (requires confirmation)
	DN	Delete a product without releasing volumes (requires confirmation)
	P	Display product details
	R	Release product volume
	S	Display complete product details
	V	Display volume details

Archive

Resource list	Line operator	Description
Data sets	C	Change data set details
	D	Delete data set (requires confirmation)
	I	Display data set details
	O	Display owner information
	V	Display volume details
	L	Display data set chain list
VRSs	C	Change VRS details
	D	Delete VRS (requires confirmation)
	L	Display VRS chain list
	M	Display matching resources
	O	Display owner information
	S	Display VRS information

All users are able to use the line operators that only display information about the resources, but only those users with read/write access to the resources are able to perform some actions against them, such as deleting or changing the information.

12.1.2 Displaying information from DFSMSrmm

You can request detailed information about any single resource defined to DFSMSrmm by using the display function in the ISPF dialog, or any of the RMM TSO **LIST** subcommands.

When using the ISPF dialog, you can reach the resource display panels by using different paths. You can also reach the display panels using the following fast path commands from the command line:

- **VOLUME DISPLAY**
- **DATASET DISPLAY**
- **RACK DISPLAY**
- **VRS DISPLAY**
- **CONTROL SYSTEM**
- **OWNER DISPLAY**

- **PRODUCT DISPLAY**

In the following sections, we include some of the most commonly used display options.

Displaying volume details

You can reach the Volume Display panel using the paths in Figure 12-30.

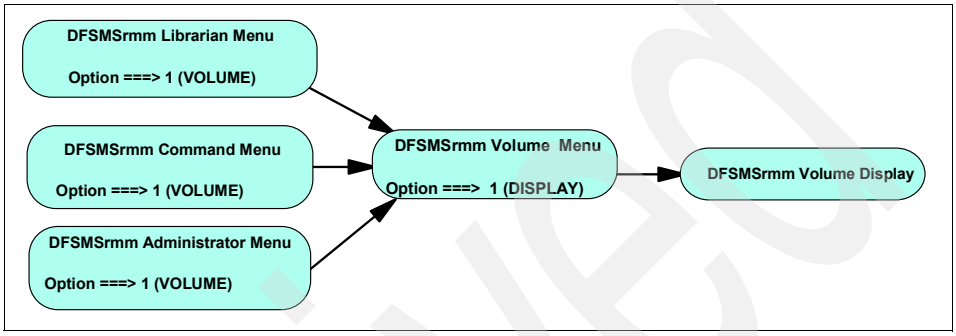


Figure 12-30 Reaching Volume Display panel

Here is an example of how to display detailed information about one volume:

1. Select option **5 (COMMANDS)** from the Removable Media Manager primary menu (Figure 12-31).

```
                    REMOVABLE MEDIA MANAGER (DFSMSrmm)
Option ===> 5

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.




5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-31 Removable Media Manager primary menu

Archived

2. In the Command Menu, select option **1 (VOLUME)** (Figure 12-32).

```
                          DFSMSrmm Command Menu
Option ===> 1

0  OPTIONS   - Specify dialog options and defaults
1  VOLUME    - Volume commands
2  RACK      - Rack and bin commands
3  DATA SET  - Data set commands
4  OWNER     - Owner commands
5  PRODUCT   - Product commands
6  VRS       - Vital record specifications
7  CONTROL   - Display system control information


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




5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-32 DFSMSrmm Command Menu

3. In the Volume Menu, select option **1 (DISPLAY)** (Figure 12-33).

```
                          DFSMSrmm Volume Menu
Option ===> 1

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.



5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-33 DFSMSrmm Volume Menu

Archived

4. In the Volume Display panel, enter a Volume serial identifier. In our case, we introduce TST010 in Figure 12-34.

```
                          DFSMSrmm Volume Display
Command ===>

Volume serial . . . TST010
```

Figure 12-34 DFSMSrmm Volume Display

5. You will see the DFSMSrmm Volume Details panel shown in Figure 12-35 and Figure 12-36.

```
                     DFSMSrmm Volume Details - TST010
Command ===>

Volume . . . . . : TST010     VOL1 volser :         Rack number : TST010
Media name . . . . : 3590                           Status  . . : MASTER
                                                                  More:   - +
Volume type  . . . : PHYSICAL       Stacked count  . . . . . . : 0
Retention date . . : 2001/061       Expiration date  . . . . . : 2001/059
Set retained . . . : NO             Original expiration date . :

Description  . . . :

Data set name  . . : 'YCJRES1.TEST1.VRS2CTLG'
Media type . . . . : HPCT           Release actions:
Label  . . . . . . : SL               Return to SCRATCH pool . : YES
  Current version  :                  Replace volume . . . . . : NO
  Required version :                  Return to owner  . . . . : NO
Density  . . . . . : IDRC             Initialize volume  . . . : NO
Recording format . : 128TRACK         Erase volume . . . . . . : NO
Compaction . . . . : YES              Notify owner . . . . . . : NO
Attributes . . . . : NONE             Expiry date ignore . . . : NO
Availability . . . : VITAL RECORD     Scratch immediate  . . . : NO

Owner  . . . . . . : YCJRES1        Owner access . . . . . . . : ALTER
Assigned date  . . : 2001/059       Assigned time  . . . . . . : 16:42:41
File 1 system ID . : SC63
Security name  . . :
Classification . . :
Account number . . :
                                    MVS use  . . . . . . . . . : YES
Jobname  . . . . . : YCJRES1F       VM use . . . . . . . . . . : NO
```

Figure 12-35 DFSMSrmm Volume Details menu (1 of 2)

Archived

```
Loan location  . . :                Last changed by  . . . . . : *HKP

Previous volume  . :                Next volume  . . . . . . . :

Volume access list :                Access . . . . . . . . . . : NONE
  User . . . . . . :                  User . . . . . . . . . . :
  User . . . . . . :                  User . . . . . . . . . . :
  User . . . . . . :                  User . . . . . . . . . . :
  User . . . . . . :                  User . . . . . . . . . . :
  User . . . . . . :                  User . . . . . . . . . . :

Volume use count . : 1              Volume usage (Kb)  . . . . : 28
Capacity (Mb)  . . : 9536           Percent full . . . . . . . : 0
Create date  . . . : 2001/053       Create time  . . . . . . . : 12:23:28
                                      System ID  . . . . . . . : SC63
Date last read . . : 2001/059       Date last written  . . . . : 2001/059

Drive last used  . : 0B91

Volume sequence  . : 1              Number of data sets  . . . : 1
                                    Data set recording . . . . : ON
Errors:
  Temporary read . : 0              Temporary write  . . . . . : 0
  Permanent read . : 0              Permanent write  . . . . . : 0

Actions pending:
  Return to SCRATCH pool . : NO     Initialize volume  . . . . : NO
  Replace volume . . . . . : NO     Erase volume . . . . . . . : NO
  Return to owner  . . . . : NO     Notify owner . . . . . . . : NO

Location . . . . . : LIB1           Destination  . . . . . . . :
Location type  . . : AUTO           In transit . . . . . . . . : NO

In container . . . :

Storage group  . . : SGLIB1         Home location  . . . . . . : LIB1
                                    Required location  . . . . :
Move mode  . . . . : AUTO           Movement tracking date . . :

Bin number . . . . :                Media name . . . . . . . . :
Old bin number . . :                Media name . . . . . . . . :

Product details:
  Product number . :
  Level  . . . . . :
  Feature code . . :
```

Figure 12-36 DFSMSrmm Volume Details menu (1 of 2)

Archived

You can get the same detailed information about a volume using the RMM TSO **LIST** command as shown in Example 12-6.

Example 12-6 TSO list volume

```
RMM LISTVOLUME TST010 ALL
```

You can see the output for this command in Figure 12-37 and Figure 12-38.

```
Volume information:
Volume = TST010    Rack   = TST010   Owner = YCJRES1   Jobname = YCJRES1F
  Type = PHYSICAL            Stacked count = 0
Create date     = 2001/053   Create time   = 12:23:28
Assigned date   = 2001/059   Assigned time = 16:42:41
Expiration date = 2001/059   Original      =
Retention date  = 2001/061   Set retained  = NO
Data set name = YCJRES1.TEST1.VRS2CTLG
Volume status:
Status = MASTER    Availability = Vital Record      Label = SL
Current label version =           Required label version =
Media information:
Density = IDRC  Type = HPCT      Format = 128TRACK  Compaction = YES
Special attributes   = NONE
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  = SGLIB1
Loan location  =          Account =
Description    =
Security class =          Description =

Access information:
Owner access = ALTER   Volume access = NONE    Last change = *HKP
VM use = N   MVS use = Y
Access list:

Statistics:
Number of data sets = 1       Data set recording= ON
Volume usage(Kb)= 28          Use count         = 1
Volume capacity = 9536        Percent full      = 0
Date last read  = 2001/059    Date last written = 2001/059
Drive last used = 0B91
Volume sequence = 1           Media name        = 3590
Previous volume =             Next volume       =
```

Figure 12-37 RMM TSO LISTVOLUME command output (1 of 2)

Archived

```
Product number  =             Level             = V  R  M
Feature code    =
Error counts:
Temporary read = 0      Temporary write = 0
Permanent read = 0      Permanent write = 0

Store information:
Location      = LIB1      Destination   =           Intransit    = N
Location type = AUTO      Home location = LIB1      Old location =
                       Required location =
Movement tracking date =                                Move mode = AUTO
Bin number     =            Media name  =
Old bin number =            Media name  =
In container   =
```

Figure 12-38 RMM TSO LISTVOLUME command output (2 of 2)

Displaying data set details

To request detailed information about a specific data set using the ISPF dialogs, you need to reach the Data Set Display panel. The following steps take you to this panel:

1. Select option **5 (COMMANDS)** from the Removable Media Manager primary menu shown in Figure 12-39.

```
                        REMOVABLE MEDIA MANAGER (DFSMSrmm)
Option ===> 5

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.




5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-39 Removable Media Manager primary panel

Archived

2. In the Librarian Menu in Figure 12-40, select option **3 (DATA SET).**

```
                              DFSMSrmm Command Menu
Option ===> 3

0  OPTIONS   - Specify dialog options and defaults
1  VOLUME    - Volume commands
2  RACK      - Rack and bin commands
3  DATA SET  - Data set commands
4  OWNER     - Owner commands
5  PRODUCT   - Product commands
6  VRS       - Vital record specifications
7  CONTROL   - Display system control information


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




5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-40 DFSMSrmm Command Menu

3. In the Data Set Menu in Figure 12-41, select option **1 (DISPLAY).**

```
                              DFSMSrmm Data Set Menu
Option ===> 1

0  OPTIONS       - Specify dialog options and defaults
1  DISPLAY       - Display data set information
2  ADD           - Add a data set
3  CHANGE        - Change data set information
4  DELETE        - Delete data set information
5  SEARCH        - Search for data sets


Enter selected option or END command. For more information, enter HELP or PF1
5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-41 DFSMSrmm Data Set Menu

4. In the Data Set Display panel in Figure 12-42, enter the Data set name and the Volume serial number of the data set for which you want to get the detailed information.

```
                          DFSMSrmm Data Set Display
Command ===>

Enter Data set name and volume serial:

Data set name  . . . . YCJRES1.DATASET.TEST1

Volume serial  . . . . TST02O

Physical file sequence number  . .           Default is 1
```

Figure 12-42 DFSMSrmm Data Set Display panel

5. You will see the Data Set Details panel shown in Figure 12-43 and Figure 12-44.

```
                          DFSMSrmm Data Set Details
Command ===>

Data set name . . : 'YCJRES1.DATASET.TEST1'
Volume serial . . : TST02O      Physical file sequence number . . : 1
Owner . . . . . . : YCJRES1     Data set sequence number  . . . . : 1
                                                                More:     +
Job name  . . . . : YCJRES1B
Step name . . . . : STEP1                    Record format . . . . : FBA
Program name  . . : IEBGENER                 Block size  . . . . . : 27951
DD name . . . . . : SYSUT2                   Logical record length : 121
Create date . . . : 2001/058    YYYY/DDD     Block count . . . . . : 1
Create time . . . : 11:46:46                 Total block count . . : 1
System id . . . . : SC63                     Percent of volume . . : 0
                                             Device number . . . . : 0B90

Last job name . . : YCJRES1B                 Last DD name  . . . . : SYSUT2
Last step name  . : STEP1                    Last device number  . : 0B90
Last program name : IEBGENER
Date last read  . : 2001/058                 VRS management value  : D99000
Date last written : 2001/058                 Management class  . . : MCDB22
                                             Data class  . . . . . :
Retention date  . : WHILECATLG               Storage class . . . . : SCLIB1
VRS retained  . . : YES                      Storage group . . . . : SGLIB1

Security name . . :
Classification  . :
Primary VRS details:
 VRS name . . . : YCJRES1.D*.**
 Job name . . . :                              VRS type  . . . . . : DSN/MC
```

Figure 12-43 DFSMSrmm Data Set Details panel (1 of 2)

Archived

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

Catalog status  . : YES
Abend while open  : NO
```

Figure 12-44 DFSMSrmm Data Set Details panel (2 of 2)

You can get the same detailed information about a data set using the RMM TSO **LIST** subcommand. Here is the command needed in our sample:

```
RMM LISTDATASET YCJRES1.DATASET.TEST1 VOLUME(TST020)
```

You can see the output for this command in Figure 12-45.

```
Data set name = YCJRES1.DATASET.TEST1
Volume        = TST020              Physical file sequence number = 1
Owner         = YCJRES1                       Data set sequence = 1
Create date   = 2001/058   Create time = 11:46:46 System ID        = SC63
Block size          = 27951      Block count          = 1
Percent of volume   = 0          Total block count    = 1
Logical Record Length = 121      Record Format        = FBA
Date last written   = 2001/058   Date last read       = 2001/058
Job name            = YCJRES1B   Last job name        = YCJRES1B
Step name           = STEP1      Last step name       = STEP1
Program name        = IEBGENER   Last program name    = IEBGENER
DD name             = SYSUT2     Last DD name         = SYSUT2
Device number       = 0B90       Last Device number   = 0B90
Management class    = MCDB22     VRS management value = D99000
Storage group       = SGLIB1     VRS retention date   = WHILECATLG
Storage class       = SCLIB1     VRS retained         = YES
Data class          =            ABEND while open     = NO
                                 Catalog status       = YES
Primary VRS details:
      Name          = YCJRES1.D*.**
      Job name      =            Type                 = DSN/MC
      Subchain NAME =            Subchain start date  =
Secondary VRS details:
      Value or class = *
      Job name      =
      Subchain NAME =            Subchain start date  =
Security Class  =           Description  =
```

Figure 12-45 RMM TSO LISTDATASET command output

You can also reach the data set information by generating a data set list, as explained in “Data set lists” on page 495, and using the line operators in this list.

Archived

Displaying shelf location details

To request detailed information about a specific shelf location using the ISPF dialogs, you must reach the Rack and Bin Display panel. Figure 12-46 shows the different ways to reach this panel.

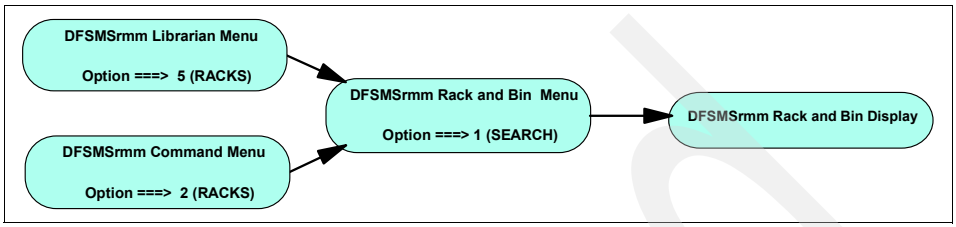


Figure 12-46 Reaching the Rack and Bin Display panel

Here is a sample of displaying detailed information of a specific shelf location.

1. Select option **5 (COMMANDS)** from the Removable Media Manager primary menu shown in Figure 12-47.

```
                        REMOVABLE MEDIA MANAGER (DFSMSrmm)
Option ===> 5

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.




5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-47 DFSMSrmm primary menu

2. In the Librarian Menu in Figure 12-48, select option **2 (RACK).**

Archived

```
                              DFSMSrmm Command Menu
Option ===> 2

0  OPTIONS   - Specify dialog options and defaults
1  VOLUME    - Volume commands
2  RACK      - Rack and bin commands
3  DATA SET  - Data set commands
4  OWNER     - Owner commands
5  PRODUCT   - Product commands
6  VRS       - Vital record specifications
7  CONTROL   - Display system control information


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




5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-48 DFSMSrmm Command Menu

3. In the Rack and Bin Menu in Figure 12-49, select option **1 (DISPLAY)**.

```
                         DFSMSrmm Rack and Bin Menu
Option ===> 1

0  OPTIONS       - Specify dialog options and defaults
1  DISPLAY       - Display racks and bins
2  ADD           - Add new racks and bins
3  DELETE        - Delete empty racks and bins
4  SEARCH        - Search for racks and bins



Enter selected option or END command. For more info., enter HELP or PF1.
5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-49 DFSMSrmm Rack and Bin Menu

4. In the Rack and Bin Display panel in Figure 12-50, enter a shelf location identifier. In our case, we entered `HGT000`.

Archived

```
                        DFSMSrmm Rack and Bin Display
Command ===>

Specify one of:
  Rack number . . . . . HGT000
    or
  Bin number  . . . . .

Location  . . . . . .            Specify a storage location to display a
                                 bin number. Leave blank to display a rack
                                 number.

Media name  . . . . . .          Required only for displaying
                                 installation-defined storage location bin
                                 numbers.
```

Figure 12-50 DFSMSrmm Rack and Bin Display

5. You will see the DFSMSrmm Rack or Bin Details panel shown in Figure 12-51.

```
                        DFSMSrmm Rack or Bin Details
Command ===>

Rack or Bin number  . : HGT000


Location  . . . . . : SHELF

Media name  . . . . . : 3490

Volume  . . . . . : HGT000

Status  . . . . . : SCRATCH
```

Figure 12-51 DFSMSrmm Rack or Bin Details

You can get the same detailed information about a volume using the RMM TSO **LIST** subcommand. The command, which is needed in our sample is shown in Example 12-7.

Example 12-7 TSO list rack

```
RMM LISTRACK HGT000
```

You can see the output for this command in Figure 12-52.

Archived

```
    Rack = HGT000  Volume = HGT000   Status = SCRATCH   Location = SHELF
    Medianame = 3490
```

Figure 12-52 RMM TSO LISTRACK command output

Displaying virtual record specifications details

As for other resources, there are different ways to reach the Display VRSs panel. Figure 12-53 shows the different paths.

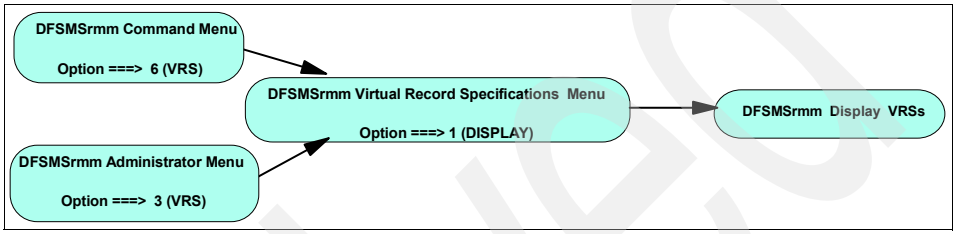


Figure 12-53 Reaching the Display VRSs panel

Here is an example of displaying details about a specific virtual record specification:

1. Select option **5 (COMMANDS)** from the Removable Media Manager initial menu shown in Figure 12-54.

```
                        REMOVABLE MEDIA MANAGER (DFSMSrmm)
Option ===> 5

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.


5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-54 DFSMSrmm Primary menu

2. In the Commands Menu shown in Figure 12-55, select option **6 (VRS)**.

Archived

```
                        DFSMSrmm Command Menu
Option ===> 6

0  OPTIONS   - Specify dialog options and defaults
1  VOLUME    - Volume commands
2  RACK      - Rack and bin commands
3  DATA SET  - Data set commands
4  OWNER     - Owner commands
5  PRODUCT   - Product commands
6  VRS       - Vital record specifications
7  CONTROL   - Display system control information


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



5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-55 DFSMSrmm Command Menu

3. In the Virtual Record Specification Menu in Figure 12-56, select option **1 (DISPLAY)**.

```
                SMSrmm Vital Record Specification Menu
Option ===> 1

0  OPTIONS   - Specify dialog options and defaults
1  DISPLAY   - Display a vital record specification
2  ADD       - Add a vital record specification
3  CHANGE    - Change a vital record specification
4  DELETE    - Delete a vital record specification
5  SEARCH    - Search for vital record specification


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



5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-56 DFSMSrmm Vital Record Specification Menu

4. In the Display Vital Record Specification panel in Figure 12-57, enter a value in one of the required fields: Data set mask, Volume serial, or VRS name. The value entered must match one of our defined VRSs. In our case, we use a data set mask for displaying the information about the VRS associated with these files.

Archived

```
                DFSMSrmm Display Vital Record Specification
Command ===>

Specify one of the following:
  Data set mask . . 'YCJRES%.T*.*'
       Job name mask . .

  Volume serial . .         ( May be generic )

 VRS name . . . .
```

Figure 12-57 Display Virtual Record Specification

5. You will get the Display Data Set VRS panel shown in Figure 12-58.

```
                        DFSMSrmm Display Data Set VRS
Command ===>

Data set mask . : 'YCJRES%.T*.*'                              GDG . : NO
     Job name mask . :

Count . . . : 2                  Retention type  . . . . . : REFDAYS
                                 While cataloged . . . . . : YES
Delay . . . : 0   Days           Until expired . . . . . . : NO

Location  . . . . . : HOME
Number in location  : 2
Priority  . . . . . : 0
                                 Release options:
Next VRS in chain . : D1O        Expiry date ignore  . . . : NO
    Chain using . . : AND        Scratch immediate . . . . : NO

Owner . . . . . : YCJRES1
Description . . :
```

Figure 12-58 DFSMSrmm Display Data Set VRS panel

You can get the same detailed information about a VRS using the RMM TSO **LIST** subcommand. The command needed in our sample is shown in Figure 12-8.

Example 12-8 TSO list DSNAME vital record specification

```
RMM LISTVRS DSNAME(YCJRES%.T*.*)
```

You can see the output for this command in Figure 12-59.

Archived

```
Data set mask = YCJRES%.T*.*                                    Type = DSNAME
Job name mask =                                Retain until expired = NO
Count        = 2     REFDAYS                Retain while cataloged = YES
Delay        = 0     Days in the HOME location
Store number = 2     REFDAYS in the HOME     location
Priority     = 0                                  Release Options:
                                               Expiry date ignore = NO
                                                Scratch immediate = NO
Next VRS in chain    = D10      using  ANDVRS

VRS Owner    = YCJRES1
Description  =
Vital Record Specification to be deleted on 1999/365
```

Figure 12-59 RMM TSO LISTVRS command output

Displaying details about DFSMSrmm options

You can use the same procedure used in the previous sections for getting information about the DFSMSrmm running options. If using the ISPF dialogs, you need to follow these steps:

1. Select option **5 (COMMANDS)** from the Removable Media Manager primary menu shown in Figure 12-60.

```
                    REMOVABLE MEDIA MANAGER (DFSMSrmm)
  Option ===> 5

  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.


  5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-60 DFSMSrmm Primary menu

2. In the Commands Menu in Figure 12-61, select option **7 (CONTROL)**.

Archived

```
                              DFSMSrmm Command Menu
Option ===> 7

0  OPTIONS   - Specify dialog options and defaults
1  VOLUME    - Volume commands
2  RACK      - Rack and bin commands
3  DATA SET  - Data set commands
4  OWNER     - Owner commands
5  PRODUCT   - Product commands
6  VRS       - Vital record specifications
7  CONTROL   - Display system control information


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




5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-61 DFSMSrmm Command Menu panel

3. In the Control Information Menu in Figure 12-62, select option **2 (SYSTEM)**.

```
                         DFSMSrmm Control Information Menu
Option ===> 2

0  OPTIONS      - Specify dialog options and defaults
1  CONTROL      - Display cds control record details
2  SYSTEM       - Display system options and defaults
3  SECURITY     - Display security classification rules
4  VLPOOLS      - Display volume pool definitions
5  MNTMSG       - Display mount message definitions
6  REJECT       - Display volumes to be rejected
7  ACTIONS      - Display volume moves and actions
8  LOCDEF       - Display location definitions


Enter selected option or END command. For more info., enter HELP or PF1.
5647-A01 (C) COPYRIGHT 1993,2000 IBM CORPORATION
```

Figure 12-62 DFSMSrmm Control Information Menu

4. You will see the System Options Display panel shown in Figure 12-63.

Archived

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

                                                                More:     +
Parmlib suffix . : PO

Operating mode . : PROTECT

Data sets:
  Control  . . . : RMM.CONTROL.DSET
  Journal  . . . : RMM.JOURNAL.DSET
  CDS id . . . . :                          Journal threshold . : 75 %
  Catalog SYSID  : NOTSET

Retention period:                           SMF:
  Default  . . . : 0                          System id . . . . : SC63
  Maximum  . . . : NOLIMIT                    Audit . . . . . . : 248
  Catalog  . . . : 6      hours               Security  . . . . : 249

PDA: ON                                     Report options:
  Block count  . : 255                        Lines per page  . : 54
  Block size . . : 27                         Date format . . . : JULIAN
  Log  . . . . . : ON
Miscellaneous:                              Auto-start procedures:
  RACF support . . . . . : NONE               Scratch procedure : EDGXPROC
  MAXHOLD limit  . . . . : 100                Backup procedure  : EDGCDSBK
  IPL check  . . . . . . : NO
  Uncatalog  . . . . . . : YES              VRS:
  System notify  . . . . : NO                 Job name  . . . . : 2
  BLP  . . . . . . . . . : RMM                Minimum count . . : 1
  Master overwrite . . . : LAST               Minimum action  . : INFO
  Message case . . . . . : MIXED              Change  . . . . . : INFO
  Uncatalog  . . . . . . : YES              VRS:
  System notify  . . . . : NO                 Job name  . . . . : 2
  BLP  . . . . . . . . . : RMM                Minimum count . . : 1
  Master overwrite . . . : LAST               Minimum action  . : INFO
  Message case . . . . . : MIXED              Change  . . . . . : INFO
  Accounting . . . . . . : JOB                VRSEL processing  : NEW
  Disposition DD name  . : DISPDD             Retain by . . . . : SET
  Disposition message ID . : EDG4054I         Move by . . . . . : SET
  TVEXT purge  . . . . . : RELEASE

SMSTAPE:
  Purge  . . . . . . . . : ASIS
  Update exits . . . . . : YES
  Update commands  . . . : YES
  Update scratch . . . . : YES
```

Figure 12-63 DFSMSrmm System Options Display panel

Archived

You can obtain the same detailed information about the system options defined in your installation using the RMM TSO **LISTCONTROL** command. The command needed in our sample is shown in Figure 12-9.

Example 12-9 TSO list CDS control information

```
RMM LISTCONTROL OPTION
```

You can see the output for this command in Figure 12-64.

```
System options:
PARMLIB Suffix  = PO
Operating mode  = P    Retention period: Default = 0       Maximum = NOLIMIT
                                         Catalog = 6     hours
Control data set name      = RMM.CONTROL.DSET
Journal file data set name = RMM.JOURNAL.DSET
Journal threshold          = 75%
Catalog SYSID   = Notset
Scratch procedure name     = EDGXPROC
Backup procedure name      = EDGCDSBK
IPL date check  = N        Date format    = J       RACF support   = N
SMF audit       = 248      SMF security   = 249     CDS id         =
MAXHOLD value   = 100      Lines per page = 54      System ID      = SC63
BLP             = RMM      TLCS V1        = N       Notify         = N
Uncatalog       = Y        VRS job name   = 2       Message case   = M
MASTER overwrite= LAST     Accounting     = J       VRS selection  = NEW
VRS change      = INFO     VRSMIN action  = INFO    VRSMIN count   = 1
Disp DD name    = DISPDD   Disp msg ID    = EDG4054I
SMSTAPE:
 Update scratch = YES      Update command = YES     Update exits   = YES
 Purge          = ASIS
Retain by       = SET      Move by        = SET
PDA: ON
 Block count    = 255      Block size     = 27      Log            = ON
TVEXT purge     = RELEASE
```

Figure 12-64 RMM TSO LISTCONTROL command output

To use the **RMM LISTCONTROL** command, you need access to the RACF profile STGADMIN.EDG.LISTCONTROL.

For more detailed information about the options you can specify in the RMM TSO **LIST** subcommands, refer to *z/OS DFSMSrmm Guide and Reference*, SC26-7404.

Archived

12.2 Reporting utilities

DFSMSrmm provides a set of reporting utilities to help the storage administrators generate useful reports about the DFSMSrmm activity and managed resources:

- EDGAUD: security and audit reports
- EDGHSKP: inventory management reports
- EDGRPTD: movement and inventory reports, and also scratch lists
- EDGRRPTE EXEC: use the extract data set as input

12.2.1 Security and audit reports

You can generate security and audit reports using the EDGAUD utility. The input for this utility is SMF records, which can be previously selected and sorted, or used as they are collected.

You must enable the DFSMSrmm SMF recording with SMFAUD or SMFSEC options in the EDGRMMxx PARMLIB member.

The EDGAUD utility reads the SMFIN file and selects records that are based on the processing criteria. The utility uses DFSORT to order the records to produce the reports you request.

Figure 12-65 contains JCL used for generating the security and audit reports.

```
//AUDREPT  EXEC PGM=EDGAUD,
//     PARM='SMFAUD(NNN),SMFSEC(NNN),SEC("SECURITY CLASSIFICATION")'
//SYSPRINT DD   program messages
//SMFIN    DD   input data set of SMF records
//AUDREPT  DD   audit report
//SECREPT  DD   security report
//SYSOUT   DD   DFSORT messages
//SYSIN    DD   select statements for audit report
```

Figure 12-65 JCL to generate security and audit reports

Security reports

Secure volumes are volumes you identify using the SECCLS PARMLIB command. The security report provides tracking information for the classified tape data you have identified.

You can use the security report to identify classified tape data sets that have been used for input or output. You can use the security report to keep track of accesses to secure volumes in your installation.

Audit reports

The audit report enables you to track changes to the control data set, identify inadvertent changes, and recover lost volumes. With EDGAUD, you can create reports that list the changes that have been made in the control data set.

For more information about the security and audit reports, refer to *z/OS V1R3.0 DFSMSrmm Reporting,* SC26-7404.

12.2.2 Inventory management reports

DFSMSrmm provides the EDGHSKP utility to help you perform inventory management. You can create standard reports as part of inventory management processing; refer to 11.1.6, “Inventory management” on page 439 for details about inventory management. These reports include the vital record specification reports, the extract data set that is used as input to report utilities, and the activity file.

- Vital record specification report

  DFSMSrmm produces a vital records retention report to the REPORT DD during inventory management processing. Use the report to:

  – Check the vital record specifications that match data sets and volumes.
  – Identify the versions of the data sets that are being retained.
  – Check the required location for each data set and volume.

- Extract data set

  You can request that an extract data set containing information from the control data set be created during DFSMSrmm inventory management.

  DFSMSrmm reads sequentially through its control data set and creates a record in the extract data set for each shelf location, volume, data set, software product, owner, and vital record specification records. Use the extract data set as input to EDGRPTD, the DFSMSrmm reporting utility and to the EDGRRPTE EXEC to create reports. The extract date set can optionally include an extended record. To request extended records, you specify `XREPTEXT DD` instead of `REPTEXT DD` when running EDGHSKP with the RPTEXT parameter.

- ACTIVITY file

  The ACTIVITY file is a pre-allocated data set, like the REPORT file. The ACTIVITY file is not intended to be a report. The ACTIVITY file contains detailed information about changes made to data sets during vital record processing.

  You can view the ACTIVITY file online. To print the ACTIVITY file, use a product such as DFSORT or DFSORT’s ICETOOL to selectively format and print fields.

Archived

You can use the DFSMSrmm-supplied sample job EDGJACTP in SAMPLIB to selectively format and print fields. This sample JCL produces reports with focus on the different types of changes that DFSMSrmm makes to data set records during inventory management.

12.2.3 Inventory, movement and scratch list reports

The DFSMSrmm utility EDGRPTD produces reports from the extract data set created using the EDGHSKP utility. It is recommended that you run storage location management before creating the extract data set to ensure that the extract data set contains the most current information about volumes that should move within the library, between the library and storage locations, or among storage locations. Figure 12-66 shows a sample JCL.

```
//RPTD     EXEC PGM=EDGRPTD,
//         PARM='SEC''(INTERNAL USE ONLY)'',DATEFORM(I),LINECOUNT(54)'
//REPTEXT  DD   DISP=SHR,DSN=HSKP.REPTEXT
//*XREPTEXT DD   DISP=SHR,DSN=HSKP.REPTEXT
//SYSPRINT DD   SYSOUT=*
//INSTVOL  DD   DISP=SHR,DSN=HSKP.INSTVOL
//INSTBIN  DD   DISP=SHR,DSN=HSKP.INSTBIN
//INSTOWN  DD   DISP=SHR,DSN=HSKP.INSTOWN
//TOSTRCK  DD   DISP=SHR,DSN=HSKP.TOSTRCK
//TOSTOWN  DD   DISP=SHR,DSN=HSKP.TOSTOWN
//FMSTBIN  DD   DISP=SHR,DSN=HSKP.FMSTBIN
//FMSTOWN  DD   DISP=SHR,DSN=HSKP.FMSTOWN
//RDYTOSCR DD   DISP=SHR,DSN=HSKP.RDYTOSCR
//SYSOUT   DD   SYSOUT=*
//SCRDATE  DD   DSN=rmm.last.run.date,DISP=OLD
//SCRLIST  DD   DISP=(,CATLG),DSN=rmm.scratch.list,
//         UNIT=SYSDA,SPACE=(TRK,(10,10)),LRECL=121,RECFM=VBA
//NEWSCR   DD   DISP=(,CATLG),DSN=rmm.new.scratch.list,
//         UNIT=SYSDA,SPACE=(TRK,(10,10)),LRECL=121,RECFM=VBA
```

Figure 12-66 Sample JCL for inventory., movement, and scratch list reporting

The DD statements you can code for inventory, movement, and scratch list reports are as follows:

INSTVOL Output file for the report containing the inventory of volumes by location sorted by volume serial number.

INSTBIN Output file for the report containing the inventory of volumes by location sorted by rack number or bin number. The storage location report is sorted by bin number. All other reports are sorted by rack number.

INSTOWN Output file for the report containing the inventory of volumes by location sorted by owner.

TOSTRCK Output file for movement reports sorted by rack number.

TOSTOWN Output file for movement reports sorted by owner.

FMSTBIN Output file for movement reports sorted by bin number.

FMSTOWN Output file for movement reports sorted by owner.

RDYTOSCR Output file for movement reports sorted in ascending order by rack report column, which can be either a bin or rack number. Volumes to be moved from locations to home locations.

NEWSCR NEWSCR is the output file for the listing of all scratch volumes returned to scratch status since the last scratch list was produced. DFSMSrmm produces the NEWSCR file when there is a valid date and time in the SCRDATE file, or the SCRDATE file contains no record.

SCRDATE This file is used to produce the scratch list report. Each time a scratch list report is produced, DFSMSrmm updates the SCRDATE file with the highest scratch date and time for a volume. DFSMSrmm uses the date and time to determine which volumes to include in the new scratch list report. DFSMSrmm includes all scratch volumes with a newer assigned date and time in the new scratch listings. You can edit the SCRDATE file, which is a single record of LRECL 80 that contains a ten-character date and an eight-character time in external format. The date format must be the same format you specified for EDGRPTD.

DFSMSrmm produces a new scratch list report only if there is a valid date or time for a volume. If there is no date or time, or the date is not valid, DFSMSrmm does not produce a new scratch list but produces a full scratch listing only in the SCRLIST DD. If the SCRDATE file is empty, the NEWSCR and SCRLIST reports are identical, and DFSMSrmm writes the highest scratch date and time to the SCRDATE file. Here is an example of the 80 byte input record:

```
01/12/200023:01:00
```

This example uses American date format. The date is 10 characters long and must start in column 1. The time is eight characters and starts in column 11. The SCRDATE file can be a new data set or an existing data set. Do not specify the date and time in the JCL using DD * because EDGRPTD updates the file with the highest scratch date and time. The SCRLIST DD can be

in any format, even a partitioned data set (PDS) member. The SCRDATE DD must be preallocated with any disposition.

SCRLIST Output file for the full scratch list report.

Note: Volumes listed in a ready-to-scratch list are excluded from the TOSTRCK or FMSTBIN reports.

You only need to code DD statements for the reports you intend to create.

There are some considerations when creating the reports:

- **Movement reports**
  You can use these reports to make an inventory of your volumes, and to identify volumes that need to be pulled and moved to other locations.

  DFSMSrmm produces reports only if there are volumes to be moved.

- **Inventory reports**
  You can use inventory reports for performing audits of your library and storage locations. To obtain the most up-to-date inventory report, move all volumes that are in transit, and confirm all moves as completed before producing the extract data set from which you produce audit reports.

  You can use the inventory reports to track logical volumes. DFSMSrmm lists all the logical and stacked volumes in the library. When you request an inventory of a VTS location, DFSMSrmm lists all the logical volumes in the library. For exported logical volumes, DFSMSrmm lists the stacked volume in the report rather than the exported logical volume.

- **Scratch lists reports**
  You can request a report that lists scratch volumes defined to DFSMSrmm. There are two types of scratch list reports:

  – The scratch list report contains the list of all volumes in scratch status.

  – The new scratch list report lists volumes returned to scratch status since the last time you ran the scratch list report.

  You can use scratch list reports to identify volumes that can be used to satisfy scratch requests.

12.2.4 DFSMSrmm-supplied EXECs

You can use the EDGJRPT sample JCL included in SAMPLIB to invoke the DFSMSrmm-supplied EXEC EDGRRPTE to create the reports listed in Table 12-2. The input to the reporting EXEC is the extract file containing extended records.

Archived

Table 12-2 DFSMSrmm reports

Report Name	Description
REPORT01	Pull list for SCRATCH tapes sorted by volume serial number
REPORT02	Pull list for SCRATCH tapes sorted by data set name
REPORT03	Inventory list by volume serial number
REPORT04	Inventory list by data set name
REPORT05	Inventory of data sets including used kilobytes
REPORT06	Inventory of volume serial numbers by location
REPORT07	Inventory of data set names by location
REPORT08	Inventory of bin numbers by location
REPORT09	List all data set names residing in a loan location
REPORT10	List all volume serial numbers residing in a loan location
REPORT11	List all multivolume and multifile sets
REPORT12	Movement report sorted by data set name
REPORT13	Movement report sorted by bin number
REPORT14	Movement report sorted by volume serial number
REPORT15	Inventory list by volume including volume count

The extract data set contains an extended extract record concatenating volume and data set information. The data set record is appended at 800 bytes. For stacked volumes, DFSMSrmm merges the stacked volume location information for all volumes contained in the stacked volume.

Creating the reports

To create the reports:

1. Create a DFSMSrmm extract data set by using the DFSMSrmm EDGHSKP utility with DD XREPTEXT. The date format is determined automatically from the header record in the extract file, and the sorting of records is based on this value.
2. Select the date format you would like for your report by altering the DATEFORM option of STEP01 in the sample EDGJRPT JCL.
3. Make sure all the messages that the DFSMSrmm subsystem issues during inventory management are copied to your job log. Refer to the step named STEP02 in the sample EDGJRPT JCL.

4. Produce the extended reports. Omit the corresponding //REPORTnn DD statements for each report that should not be created. Refer to the step named EXTRPDT in the sample EDGJRPT JCL.

Tailoring the sample JCL for EDGJRPT

Before you can use the JCL, you must customize the sample JCL EDGJRPT for your environment:

1. Modify the PAGEDEF and FORMDEF definitions in the OUTDDQ DD in step EXTRPDT.
   Specify a valid font for your printer.
   Define a printer address and a node to print your reports:
   a. Change the data set name of the MESSAGE DD statements to your own data set name of the MESSAGE file.
   b. Change the data set name of the extended extract data set to your own naming convention. You must make this change whenever the RMM.EXTRACT.FILE file reports name is specified.
   c. Change the SPACE and UNIT parameter for the SORTOUT and SYSIN statements. Calculate DASD space requirements by multiplying the number of data set records by 1400 bytes for each record.
   d. The lines per page are defined as a parameter to the EDGRRPTE REXX procedure. If you need to use a value other than 54, replace the 054 value with your own choice.
   e. Select reports by using the REPORTxx DD statements specified in the job step running the EDGRRPTE procedure. In Figure 12-67, all reports except REPORT06 and REPORT07 are selected.

Archived

```
//REPORT01 DD  SYSOUT=*,RECFM=VBA
//REPORT02 DD  SYSOUT=*,RECFM=VBA
//REPORT03 DD  SYSOUT=*,RECFM=VBA
//REPORT04 DD  SYSOUT=*,RECFM=VBA
//REPORT05 DD  SYSOUT=*,RECFM=VBA
//*EPORT06 DD  SYSOUT=*,RECFM=VBA
//*EPORT07 DD  SYSOUT=*,RECFM=VBA
//REPORT08 DD  SYSOUT=*,RECFM=VBA
//REPORT09 DD  SYSOUT=*,RECFM=VBA
//REPORT10 DD  SYSOUT=*,RECFM=VBA
//REPORT11 DD  SYSOUT=*,RECFM=VBA
//REPORT12 DD  SYSOUT=*,RECFM=VBA
//REPORT13 DD  SYSOUT=*,RECFM=VBA
//REPORT14 DD  SYSOUT=*,RECFM=VBA
//REPORT15 DD  SYSOUT=*,RECFM=VBA
//SYSTSIN  DD  *
  EX 'SYS1.SEDGEXE1(EDGRRPTE)'  -
     '054 INTERNAL USE ONLY'
```

Figure 12-67 Report selection

2. Figure 12-68 shows you how to replace the default security heading text. The security heading text can be up to 30 characters. The text can contain blanks or special characters and will be written on each page. Remember to use the correct continuation character (+) to suppress all the leading blanks in the new line.

```
  EX 'SYS1.SEDGEXE1(EDGRRPTE)'  -
     '054 INTERNAL USE ONLY'
          ################ ---- SECURITY HEADING TEXT - UP TO 30 CHARS
```

Figure 12-68 Creating a report security header

3. Optionally, add the CCARD DD to overwrite the internal SORT statements, the security header, or the lines per page. Figure 12-69 shows an example of how to specify the CCARD DD statements. Valid parameters that can be specified are:

SORTnn	(nn = 01 to 15) for the SORT statements
HEAD	for the security header
LINES	for the lines per page
/*	for the end of the records

Archived

```
//CARD     DD   *
SORT06     SORT FIELDS=(156,8,CH,A,9,6,CH,D,915,4,CH,A)
SORT06     INCLUDE COND=(5,1,CH,EQ,C'X',
SORT06             AND,(583,1,CH,EQ,C'S',
SORT06             OR,583,1,CH,EQ,C'U'))
SORT06     OPTION VLSHRT
LINES      20
HEAD       INTERNAL USE
/*
```

Figure 12-69 Defining a CCARD

Tailoring the DFSMSrmm-supplied EXEC

You must tailor the DFSMSrmm-supplied EXEC TO create your own reports.

The DFSMSrmm-supplied report REPORT01, when used as is, produces a pull-down list for scratch volumes sorted by volume serial number. Follow these steps to tailor the REPORT01 to provide information about volumes with temporary write errors instead of a pull-down list for scratch tapes:

1. Make a copy of the EDGRRPTE REXX EXEC to avoid losing any modifications you make to the DFSMSrmm-supplied reports, because you will lose your changes when DFSMSrmm replaces them.
2. To change the sort order and criteria, change SORT FIELDS and the INCLUDE statement for the REPORT01 in the EDGRRPTE REXX EXEC. To find the sort statement for the REPORT01, do a search for SORT01. You can find the fields for the SORT FIELD and the INCLUDE statement by looking at the mapping of the extended extract record EDGRXEXT.

   A simpler way to change the sort options for a report is to use the `CCARD DD` and specify all the sort options using the SORT01 statements. If you add the SYMNAMES DD statement and point to the extract data set symbols, you can simplify the specification of the sort statements:

   ```
   //SYMNAMES DD DISP=SHR,DSN=SYS1.MACLIB(EDGEXTSY)
   ```

   Figure 12-70 shows the original JCL where REPORT01 is sorted by volume serial number and volume status in the EDGRRPTE REXX.

```
sort01.1 = '  SORT FIELDS=(5,6,CH,A)              '
sort01.1 = left(sort01.1,80)
sort01.2 = '  INCLUDE  COND=(5,1,CH,EQ,C'X',      '
sort01.2 = left(sort01.2,80)
sort01.3 = '           AND,322,8,CH,EQ,C'SCRATCH ','
sort01.3 = left(sort01.3,80)
sort01.4 = '           AND,915,4,CH,LT,C'   2')    '
sort01.4 = left(sort01.4,80)
sort01.5 = '  OPTION VLSHRT                        '
sort01.5 = left(sort01.5,80)
s01 = 5
```

Figure 12-70 Sorting by volume serial number and volume status

Figure 12-71 shows 375 in the SORT FIELDS , which is the offset in the EDGRXEXT mapping macro for the temporary write errors plus the value 4 for the record length field.

```
sort01.1 = '  SORT FIELDS=(375,6,CH,A)            '
sort01.1 = left(sort01.1,80)
sort01.2 = '  INCLUDE  COND=(5,1,CH,EQ,C'X',      '
sort01.2 = left(sort01.2,80)
sort01.3 = '           AND,322,8,CH,EQ,C'SCRATCH ','
sort01.3 = left(sort01.3,80)
sort01.4 = '           AND,915,4,CH,LT,C'   2')    '
sort01.4 = left(sort01.4,80)
sort01.5 = '           AND,375,4,CH,GT,C'   0')    '
sort01.5 = left(sort01.5,80)
sort01.6 = '  OPTION VLSHRT                        '
sort01.6 = left(sort01.6,80)
s01 = 6
```

Figure 12-71 Sorting by volume serial number, status, and temporary errors

To change the report header, modify the DFSMSrmm-supplied EDGRRPTE REXX EXEC as shown in Figure 12-72.

```
t2.1 = center('Scratch Tapes by Volume Serial Number',69)
t0.2 = left('EDGRPT01',8)
```

Figure 12-72 REPORT01 report header

Figure 12-73 shows the change to create a new report header named Volumes with Temporary Errors.

```
t2.1 = center('Volumes with Temporary Errors',69)
t0.2 = left('EDGRPT01',8)
```

Figure 12-73 REPORT01 report header modified

To change the titles on the columns, modify the DFSMSrmm-supplied EDGRRPTE REXX EXEC. The title line for the report columns is `out.cs = asa.h`. You can find the definition for the variables in the sample EDGRRPTE EXEC starting at the label `const`. Figure 12-74 shows the report column headings as they are defined in the sample EDGRRPTE EXEC.

```
do h = 1 to 3
  cs     = cs + 1
  out.cs = asa.h tvolser.h tdsname.h tvolseq.h tdsnseq.h,
                 tcrdate.h texpdto.h,
                 tflag.h tltyp.h,
                 tmedty.h tmedrec.h,
                 thome.h tstore.h tloc.h,
                 terror.h
end
```

Figure 12-74 REPORT01 column headings

Figure 12-75 shows the variable `twrte.1`, which is the column heading for temporary errors.

```
do h = 1 to 3
  cs     = cs + 1
  out.cs = asa.h tvolser.h tdsname.h tcrdate.h ttwrte.h,
                 texpdto.h,
                 tflag.h tltyp.h,
                 tmedty.h tmedrec.h,
                 thome.h tstore.h tloc.h,
                 terror.h
end
```

Figure 12-75 REPORT01 column headings modified

3. To obtain the correct output, modify the DFSMSrmm-supplied EDGRRPTE REXX EXEC by specifying the appropriate output variable. You can find the definition for these variables in the sample EDGRRPTE EXEC starting at the label lclexmap. The `out.cs = asa.0` is the output value returned in the report. Figure 12-76 shows the returned value from the sample EDGRRPTE EXEC.

Archived

```
out.cs = asa.0 rvvolser rddsaname rvvolseq rddsnseq,
               rvcrdate rvexpdto,
               lclflag rvlabel,
               rvmedty rvmedrec,
               rvhloc rvloctyp lcloc lclerror
```

Figure 12-76 REPORT01 returned values

Figure 12-77 shows the addition of the `rvtwerr` variable to obtain the temporary write error information.

```
out.cs = asa.0 rvvolser rddsaname rvcrdate rvtwerr,
               rvexpdto,
               lclflag rvlabel,
               rvmedty rvmedrec,
               rvhloc rvloctyp lcloc,
               lclerror
```

Figure 12-77 REPORT01 returned values modified

4. Submit the job.

List of DFSMSrmm-supplied reports

Here you can find information about the changes introduced in the DFSMSrmm-supplied reports. You may refer to *DFSMSrmm Reporting* for more detailed information about a specific report format or field.

The general change in all the reports is a new line in the header including the time the report was created. You can see an example of the new header in Figure 12-78:

- **REPORT01: Pull-down list for SCRATCH tapes sorted by volume serial number**

  REPORT01 includes volumes in SCRATCH status and only the first file on the volume. This report is sorted by volume serial number.

Archived

```
DFSMSrmm    Internal use only              Scratch Tapes by Volume Serial Number                          PAGE -        1
EDGRPT01                                                                                                   DATE - 2000/341
                                                                                          TIME - 22:38:55
Volume                                          Vol- DSN- Create     Org. Exp.  V LBL Media    Rec. Home     S Location Sum.
Serial Data Set Name                            Seq. Seq. Date       Date       F Typ Type     Fmt  Location S Name     Error
------ ---------------------------------------- ---- ---- ---------- ---------- - --- -------- ---- -------- - -------- -----
A00031                                           1        12/05/1999              SL     *     *    SHELF                0
A00032                                           1        12/05/1999              SL     *     *    SHELF                0
A00033                                           1        12/05/1999              SL     *     *    SHELF                0
A00034                                           1        12/05/1999              SL     *     *    SHELF                0
A00035                                           1        12/05/1999              SL     *     *    SHELF                0
A00036                                           1        12/05/1999              SL     *     *    SHELF                0
A00037                                           1        12/05/1999              SL     *     *    SHELF                0
A00038                                           1        12/05/1999              SL     *     *    SHELF                0
A00039                                           1        12/05/1999              SL     *     *    SHELF                0
A00040                                           1        12/05/1999              SL     *     *    SHELF                0
A00101                                           1        12/05/1999              SL     *     *    SHELF                0
                                        End of Report. 11 Entries listed
```

Figure 12-78 Sample REPORT01 output with new header

- **REPORT02**: Pull-down list for SCRATCH tapes sorted by data set name
  REPORT02 includes volumes in SCRATCH status and only the first file on the volume. REPORT02 is sorted by data set name and volume serial number.
- **REPORT03:** Inventory list by volume serial number
  REPORT03 includes all data sets. REPORT03 is sorted by volume serial number and data set sequence number.
- **REPORT04:** Inventory list by data set name
  REPORT04 includes data sets and excludes all volumes without any data set information. REPORT04 is sorted by data set name, create date, and create time.
- **REPORT05:** Inventory of data sets including used kilobytes
  REPORT05 includes data sets and excludes all volumes without any data set information. REPORT05 is sorted by data set name, create date, and create time.
- **REPORT06:** Inventory of volume serial numbers by location
  REPORT06 includes all volumes residing in one of the three built-in storage locations or installation-defined storage locations. REPORT06 is sorted by storage location, volume serial number, and data set sequence number.
- **REPORT07:** Inventory od data set names by location
  REPORT07 includes all volumes residing in one of the DFSMSrmm-supplied built-in storage locations or installation-defined storage locations. REPORT07 is sorted by storage location, data set name, create data, and create time.
- **REPORT08:** Inventory of bin numbers by location
  REPORT08 includes all volumes residing in one of the three built-in storage locations or installation-defined storage locations. REPORT08 is sorted by storage location, bin number, data stored, and data set name.

Archive

- **REPORT09:** List all data set names residing in a loan location
  REPORT09 includes all volumes residing in a LOAN location. REPORT09 is sorted by loan location, data set name, create data, and create time.
- **REPORT10:** List all volume serial numbers residing in a loan location
  REPORT10 includes all volumes residing in a loan location. REPORT10 is sorted by loan location, volume serial number, and data set sequence number.
- **REPORT11:** List all multivolume an multifile sets
  REPORT11 includes all multifile volumes and multivolume files. REPORT11 is sorted by the first file on the first volume of the multivolume or multifile set, multi data set multivolume token, volume sequence number, and data set sequence number. There is a new column in the report:
  Vol-Cnt. The volume count.

  You can see an example of the new report in Figure 12-79.

```
DFSMSrmm    Internal use only              Multi-Volume/Multi-Data Set Report                                  PAGE -        1
EDGRPT11                                                                                                       DATE - 2000/341
                                                                                                                 TIME - 22:38:55
Volume Vol- Vol- DSN-                                                Expiration First  Prev.  Next   Create   Creating Create     Create
Serial Seq. Cnt. Seq. Data Set Name                                  Date       Volser Volser Volser Userid   Jobname  Date       Time
------ ---- ---- ---- ------------------------------------------- ---------- ------ ------ ------ -------- -------- ---------- ------
SC0464    1    1    1 ADDONS.PANELS                                          SC0464                SIEGEL            1992/240   084146
SC0464    1    1    2 ADDONS.SKELS                                           SC0464                SIEGEL            1992/240   084154
SC0464    1    1    3 ADDONS.EXEC                                            SC0464                SIEGEL            1992/240   084159
SC0464    1    1    4 ADDONS.EXECFB                                          SC0464                SIEGEL            1992/240   084205
SC0464    1    1    5 ADDONS.MSGS                                            SC0464                SIEGEL            1992/240   084218
SC0464    1    1    6 ADDONS.INITVARS                                        SC0464                SIEGEL            1992/240   084223
SC0464    1    1    7 ADDONS.LOAD                                            SC0464                SIEGEL            1992/240   084226
SC0464    1    1    8 ADDONS.CNTL                                            SC0464                SIEGEL            1992/240   084232
SC0464    1    1    9 ADDONS.SOURCE                                          SC0464                SIEGEL            1992/240   084238
SC0464    1    1   10 ADDONS.OBJ                                             SC0464                SIEGEL            1992/240   084248
                                  Used kilobytes for volume SC0464  2073629
SC0695    1    5    1 BSYSMF.WEEK.G9519V00                          1995/315   SC0694        SC0702 STCOPC   SMFWEEK2 1992/240   140404
SC0702    2    5    1 BSYSMF.WEEK.G9519V00                          1995/315   SC0694 SC0695 SC0699 STCOPC   SMFWEEK2 1992/240   140646
SC0699    3    5    1 BSYSMF.WEEK.G9519V00                          1995/315   SC0694 SC0702 SC0704 STCOPC   SMFWEEK2 1992/240   140934
SC0704    4    5    1 BSYSMF.WEEK.G9519V00                          1995/315   SC0694 SC0699 SC0706 STCOPC   SMFWEEK2 1994/207   141217
SC0706    5    5    1 BSYSMF.WEEK.G9519V00                          1995/315   SC0694 SC0704        STCOPC   SMFWEEK2 1992/240   141925
                                  Used kilobytes for volume SC0465    73629
....
                                       End of Report.   5383 Entries listed
```

Figure 12-79 Sample REPORT11 output with new Vol-Cnt. column

- **REPORT12:** Movement report sorted by data set name
  REPORT12 includes all volumes moving among the three built-in storage locations or installation-defined storage locations. REPORT12 is sorted by destination, storage location, data set name, create date, and create time.
- **REPORT13:** Movement report sorted by bin number
  REPORT13 includes data set information. The report is sorted by destination, location, and bin number.

- **REPORT14:** Movement report sorted by volume serial number
  REPORT14 includes data sets. The reports is sorted by destination, location, and volume serial number.
- **REPORT15:** Inventory list by volume including volume count
  REPORT15 provides a count of the maximum number of tapes in a multivolume chain. If a volume is not part of a multivolume chain, the count is set to 1. REPORT15 is sorted by volume serial number and data sequence number.

12.3 DFSORT reporting

You can create reports from the data in DFSMSrmm extract data set, activity report, and SMF records using DFSORT's multipurpose ICETOOL utility.

DFSMSrmm provides sample jobs that use DFSORT, often through ICETOOL to produce sample reports. These sample jobs are included in SYS1.SAMPLIB.

You can use the DFSMSrmm-supplied samples without modification, or use them as examples to produce specific customized reports from DFSMSrmm information. You can change the DFSORT or ICETOOL control statements and job steps to create reports for your installation.

You can use DFSORT symbols in ICETOOL and DFSORT jobs to create reports for DFSMSrmm-managed resources. DFSORT symbols can increase your productivity by automatically providing the positions, lengths, and formats of the fields and the values of the constants associated with particular DFSMSrmm data you are processing with ICETOOL and DFSORT.

Symbols can help standardize your DFSORT applications and increase your productivity. You can use a symbol anywhere you can use a field or constant in any DFSORT control statement or ICETOOL operator. DFSORT symbols can be up to 50 characters, are case-sensitive, and can include underscore characters.

To use symbols with DFSORT and ICETOOL jobs, you just:

1. Create or obtain DFSORT symbol data sets that describe the data you want to process. Symbol data sets contain symbols that map the fields in your records, and constants used for comparisons, titles, headings, and so on. The symbols are specified in DFSORT's simple but flexible SYMNAMES statement format. You can easily add, delete, or modify symbols using an editor, such as ISPF EDIT.
2. Include a SYMNAMES DD statement specifying the symbol data sets you want to use. You can use SYMNAMES to specify one symbol data set or many concatenated symbol data sets.

3. Use the symbols from SYMNAMES in DFSORT control statements and ICETOOL operators. You can mix symbols (for example, Last_Name) with regular fields (for example, 20,5,CH) and constants (for example, C'Yaeger') for processing with ICETOOL and DFSORT.

DFSMSrmm provides you with symbols you can use in DFSORT and ICETOOL jobs to create reports for DFSMSrmm-managed resources. These symbol mappings are available in SYS1.MACLIB after SMP/E APPLY processing, as members EDGACTSY, EDGEXTSY, and EDGSMFSY. You can access these symbols in your DFSORT and ICETOOL jobs by pointing the SYMNAMES DD statement directly to any of these members.

For releases prior to OS/390 R10, you can obtain the DFSMSrmm symbols mapping and samples from:

`http://www.ibm.com/storage/dfsort/`

For more information about using DFSORT symbols to obtain customized reports, refer to *z/OS V1R3.0 DFSMSrmm Reporting,* SC26-7404. If you need more details about using symbols with DFSORT, refer to *Getting Started with DFSORT Release 14,* SC26-4109.

12.4 DFSMSrmm report generator

The DFSMSrmm report generator is an easy to use ISPF application that allows you to create reports. With the report generator, you can create reports using sequential data sets as input. You must specify mappings of the records in the input data set for the report generator to be able to pick out information in the input data set. You can use EDGHSKP utility to create the extract data set and use it as input file for the report generator. If you use the extract data set as input, then DFSMSrmm provides mapping macros for the records in the extract data set. DFSMSrmm provides report types that associate the input file with the needed mapping macros.

To create a report with the report generator you need to define three variables:

- Report type: In this variable you select the type of report you want to generate, for example, volume lists, data set lists, movement lists, and some more. DFSMSrmm provides report types for the extract file, SMF records, and ACTIVITY file.
- Report definition: Allows you to select the fields you want to see in the reports you generate. Also, you can select the criteria fields for the reports.

- Reporting tool: You can select the reporting tool you want to use. The default reporting tool is DFSORT ICETOOL. You can set up the report generator for other reporting tools if you have REXX programming skills.

The report generator guides you through ISPF panels to define your reports and then it produces a JCL. In the generated JCL are included all the commands needed for producing the report as you have selected. To obtain the report, you must submit the JCL. The JCL is saved to be reused for subsequent reports.

The report generator helps you generate JCL that must be run as a batch job to generate a report.

12.4.1 Setting up the report generator

You need to define up to three report definition libraries:

1. The *product* library.

   This is the library that ships with the product.

   The default is SYS1.SAMPLIB.

2. The *installation* library.

   This is a library the system programmer or storage administrator will provide for the whole installation.

   When the system programmer or librarian creates reports for an installation, they need to specify the installation library name as their *user* library since updates are only made available in the library defined as the *user* library. Once the report types, definitions and tools have been customized as required in your installation, the system programmer uses his user library as the installation library.

   The default is empty.

3. The *user* library.

   This library can be specified by the user for his own reports. Updates are only performed in the user library.

   The default is userid.REPORT.LIB.

Also, a JCL library can be specified where to save the generated report JCL. The default is userid.REPORT.JCL. The default libraries are defined in the EDGRMAIN EXEC, refer to “Implementation steps” on page 543 for more information.

All four libraries must be partitioned data sets with fixed 80 byte records. If they are not predefined, DFSMSrmm will allocate them automatically with a primary and secondary space of 10 tracks and 50 directory blocks. All data sets need to

Archived

be specified like in ISPF: fully qualified with single quotes, or without quotes and HLQ. The data set names are automatically expanded to the full qualifier including the single quotes. If the user has specified NOPREFIX in the ISPF profile, then the RACF user ID will be used as the HLQ.

You can select from predefined report types and report definitions or create your own report types and definitions.

To specify the four libraries needed for the report generator, you can use the ISPF panels. The library names are initially set to the values defined in the EDGRMAIN EXEC. You can change the names if you want.

If you select option **0 (OPTIONS)** in the Removable Media Manager primary menu, then you receive the Dialog Options Menu panel, as shown in Figure 12-80.

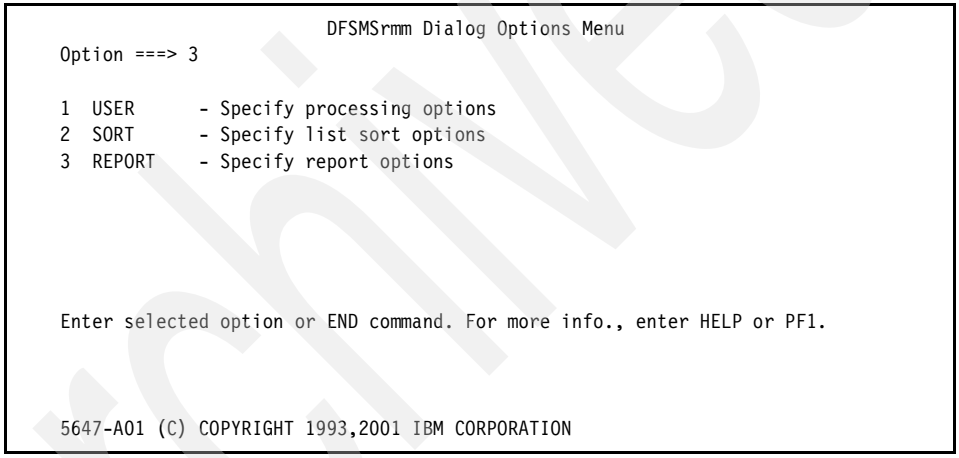


Figure 12-80 Dialog Options Menu panel

From this panel, select option **3 (REPORT),** and you will see the Report Options panel; see Figure 12-81. In this panel, you enter the name of the four libraries required.

Archived

```
                          DFSMSrmm Report Options
Option ===>

Report definition libraries:
   User. . . . . . . . . . 'YCJRES1.REPDEF'
   Installation  . . . . . 'RMM.SAMPLE.REPORTS'
   Product . . . . . . . . 'SYS1.SAMPLIB'

User report JCL library  . 'YCJRES1.REPJCL'



DFSMSrmm allocates user libraries if they do not exist.


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

Figure 12-81 DFSMSrmm Report Options panel

Implementation steps

There are a number of steps to follow to be able to use the report generator, and these steps will vary depending on the type of user.

Storage administrator

1. Define the installation library to be used in your installation as user library in the report options panel and allocate it manually or automatically.
2. Define the JCL library and specify the product library. The product library by default is SYS1.SAMPLIB.
3. Provide READ authority to the necessary users to the installation and product libraries.
4. Select the report types panel and add or change reports types shipped with the product, and set them up for your users.
5. Select the report definition panel and add or change the reports shipped with the product, and set them up for your users.
6. Define the installation library name in EXEC EDGRMAIN as well as the product library name, if not equal to SYS1.SAMPLIB.
7. Update the default naming convention in EXEC EDGRMAIN for the user library name and the JCL library name if necessary.

Librarian and general user

1. Verify the user library name in the report options panel and allocate it manually or automatically.

Archived

2. Verify the JCL library and specify the product and installation library. The names may be provided by the storage administrator if not filled already.
3. Select the report types panel and add or change report types shipped with the product or installation support if needed.
4. Select the report definition panel and add or change reports shipped with the product or installation support if needed.
5. Fill out the job card in the DFSMSrmm options panel.
6. Create the report JCL to be submitted and get the report.

12.4.2 The report type support

The report type contains the basic information you need for identifying the records for reporting.

The report type for the DFSMSrmm specific data sets is provided in SYS1.SAMPLIB(EDGGRTD) with the product, and can be copied and changed by the proper person. This member name EDGGRTD is a reserved name and it is used in all three libraries (product, installation, and user) to store report types. Updates to EDGGRTD member are only performed in the user library. The report generator application reads the information from all three libraries, if defined, starting with the user library, installation library, and the product library. The information is presented in an ISPF table to the user. DFSMSrmm does not display duplicate entries in the ISPF table. DFSMSrmm stores added and changed entries in EDGGRTD in the user library.

All the changes on entries from the installation and product library are only performed in the ISPF table, not in the libraries itself. On exit from the ISPF table, all entries being added, changed, or read from the user library are written to the user library member.

Report types are only used when a new report definition is created. Once the report definition has been created, all report type information is copied to the report definition. Changes in the report types will not be reflected in existing report definitions.

The report type contains the following information:

- The report type name

  This is the unique identifier of all the report types.
- The report type description

  This is a required field.

- The name of the associated macros

  If a record is mapped by the concatenation of more than one macro (for example, DFSMSrmm SMF records have two macros; EDGSMFAR for the SMF header; and EDGSVREC for DFSMSrmm SMF volume record), then you can specify up to five macros. The macro structures are concatenated in the sequence they are specified. Macros must be defined in ASSEMBLER.

  One macro name is required.

- The name of the macro library

  All macros need to be defined in the same library.

  This is a required field.

- The report input data set

  This is the data set containing the input records for the report mapped by the macro definition. If you are using GDGs or always use the same data set name, the data set name can already be specified here. If you plan to use different input data set names, then you should leave this field empty, since you will be prompted for the name in any case before the report JCL is created. The input data set must be a sequential file.

  This is an optional field.

- The report creation information

  This is the creating timestamp and user ID.

  This information is created automatically.

- The report last change information

  This is the last change timestamp and user ID.

  This information is created automatically.

- The record selection criteria

  The record selection criteria specify the criteria for selecting records from the input data set in case different record types exist. Similar to the selection of records in the report definition dialog, you can specify criteria for the different fields. Refer to 12.4.4, “The report definition support” on page 547 for more details about the selection criteria. This criteria is not displayed in the ISPF table. You have to select the entry and then you can specify the criteria. The selection criteria is passed to the report definition, and can be changed there as well.

12.4.3 The reporting tool support

The report generator allows the use of different reporting tools. The selection of the fields and criteria in the report definition is independent from the used

reporting tool. The intention of the report generator is to create a JCL that can be executed in batch. No support for reports running in foreground is provided. The creation of the report JCL requires an EXEC member with the code to read the report definition member and to write the report JCL. Optionally, a skeleton library can be used. With this approach you can create your own report JCL for any type of reporting tool (for example, SAS or SYNCTOOL). The list of reporting tools is provided in SYS1.SAMPLIB(EDGGTOOL) with the product. The reporting tools list can be modified and stored in the installation library for the entire installation, or stored in a user library for private use. The member EDGGTOOL is a reserved name and is used in all three libraries similar to the report type member EDGGRTD. The reporting tool definitions are presented in an ISPF table to the user. Duplicate entries are ignored. Added or changed entries are stored in EDGGTOOL in the user library.

The report generator copies information about the contents of the product, installation, and user libraries into an ISPF table that remains for a session. When the user requests that a library member is deleted, the report generator deletes the entry from the ISPF table, and not the installation or product library. The changes that the user makes to the ISPF table are reflected in the user library member.

The report generator provides support for DFSORT ICETOOL.

The reporting tool EXEC needs to be in a library defined in SYSPROC or SYSEXEC.

The reporting tool definition contains the following information:

- The reporting tool EXEC name
  The reporting tool EXEC is the name of the member that creates the reporting JCL out of the report definition.
  This is a unique identifier of all reporting tools.
- The reporting tool description
  The reporting tool description is the synonym of the reporting tool (for example, ICETOOL, SYNCTOOL, SAS).
  This is a required field.
- The column space
  This is the number of spaces between the report columns. The value depends on the reporting tool being used. It will only be used for the calculation of the report width.
  This is a required field.
- The report creation information
  This is the creating timestamp and user ID.
  This information is created automatically.

- The report last change information
  This is the last change timestamp and user ID.
  This information is created automatically.

12.4.4 The report definition support

The report definition allows the definition of individual reports. That includes the report header and footer, the fields, the criteria, the sort order and the grouping of records. Each report definition is stored in a separate member in one of the three report libraries (user, installation, product). Therefore, the report definition name can only have eight characters. The names EDGGRTD and EDGGTOOL are reserved names and cannot be used for report definitions. For the creation of report definitions a report type and reporting tool is required. The information about the used macro, the macro library and the record selection criteria specified and copied from the report type is visible to the user and can be changed. The reporting tool can be changed at any time.

12.4.5 Using the report generator

The ISPF dialog has been changed to include the new report generator options. The new REPORT option is included in the following panels:

- Dialog Options Menu panel (Figure 12-80 on page 542)
- Command Menu panel (Figure 12-82)

```
                          DFSMSrmm Command Menu
Option ===>

0  OPTIONS   - Specify dialog options and defaults
1  VOLUME    - Volume commands
2  RACK      - Rack and bin commands
3  DATA SET  - Data set commands
4  OWNER     - Owner commands
5  PRODUCT   - Product commands
6  VRS       - Vital record specifications
7  CONTROL   - Display system control information
8  REPORT    - Report generator


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

5647-A01 (C) COPYRIGHT 1993,2001 IBM CORPORATION
```

Figure 12-82 Command Menu panel with REPORT option

- User Menu panel (Figure 12-83)

```
                              DFSMSrmm User Menu
Option ===>

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

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


5647-A01 (C) COPYRIGHT 1993,2001 IBM CORPORATION
```

Figure 12-83 User Menu panel with REPORT option

- Librarian Menu panel (Figure 12-84)

```
                           DFSMSrmm Librarian Menu
Option ===>

0  OPTIONS    - Specify dialog options and defaults
1  VOLUME     - Display or change volume information
2  ADDPP      - Add a product
3  PRODUCTS   - Search for products
4  OWNER      - Display or change owner information
5  RACKS      - Manipulate library racks and storage location bins
6  ADDVOL     - Add a volume to the library
7  SCRATCH    - Add SCRATCH volumes to the library
8  RELEASE    - Release volumes
9  CONFIRM    - Confirm librarian/operator actions
A  REQUEST    - Assign a user volume to an owner
B  STACKED    - Add stacked volumes
C  REPORT     - Work with reports


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

Figure 12-84 Librarian Menu panel with REPORT option

From the User, Command, and Librarian Menu panels, you can reach the report generator panel (Figure 12-85).

Archived

```
                          DFSMSrmm Report Generator
Option ===>

0  OPTIONS        - Specify dialog options and defaults
1  REPORT         - Work with reports
2  REPORT TYPE    - Work with report types
3  REPORTING TOOL - Work with reporting tools


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


5647-A01 (C) COPYRIGHT 1993,2001 IBM CORPORATION
```

Figure 12-85 Report Generator panel

From the Dialog Options Menu panel (Figure 12-80 on page 542), selecting option **3 (REPORT)**, and from the Report Generator panel (Figure 12-85), selecting option **0 (OPTIONS),** you can reach the Report Options panel, shown in Figure 12-81 on page 543. In this panel, you can change the names of the different libraries needed for the report generator.

Working with report definitions

In the Report Generator panel (Figure 12-85), select option **1 (REPORT)**. You will then reach the Report Definition Search panel (Figure 12-86).

```
                        DFSMSrmm Report Definition Search
Command ===>

Report name . .           May be generic. Leave blanck for all reports .

User ID . . . .           Leave blank for all user IDs.

Libraries (enter S):      Select one or more library.
  _User                  Default are all defined libraries.
  _Installation
  _Product



The following line commands are available when the list is displayed:
  A - Add a new report definition   D - Delete a report definition
  G - Generate and save the JCL     J - Edit and submit the JCL
  N - Copy a report definition      S - Display or change the report def.  T -
Select a reporting tool
```

Figure 12-86 Report Definition Search panel

From this panel, you can enter a Report name to search for report definitions by name. The report definition name is the name of a member in one of the report definition libraries. You can also use the User ID field to search for report definitions updated by a specific user. Finally, you can select one or more of the user, installation or product Libraries among the three specified. You can also leave these fields blank to get a list of all available report definitions. In Figure 12-87, you can see an example Report Definitions panel with a list of available reports.

```
                        DFSMSrmm Report Definitions          Row 1 to 3 of 3
Command ===>

The following line commands are valid: A,D,G,J,N,S and T

S Name     Report title                  Report type                User id
- -------- ----------------------------- -------------------------- -------
  SCRVOL   Scratch Volume Inventory      Inventory of volumes       YCJRES1
  MULTIDSN MV/MF Inventory               Extended Extract File      YCJRES1
  VOLAUD   SMF Audit Report per Volume   SMF Audit of Volumes       YCJRES1
******************************** Bottom of data ********************************
```

Figure 12-87 Report Definitions panel

The available line commands are:

A	Add a new report definition
D	Delete a report definition
G	Generate and save a report JCL
J	Edit and submit a report JCL
N	Copy a report definition
S	Display or change a report definition
T	Select a reporting tool

Creating a report definition

You can create a new report definition selecting the option **A** in the Report Definitions panel. You can also create a new report definition by copying an existing one, by selecting option **N** in the Report Definitions panel. In both cases you receive the screen in Figure 12-88 prompting for a new report name.

```
Enter the report name . . . .
```

Figure 12-88 New report name prompt

If you select option **A**, then you must select the report type (Figure 12-89) and the reporting tool to use in the new report definition (Figure 12-90).

```
                    Select Report Type                    Row 1 to 7 of 7

S Report type                    Name
- ------------------------------ --------
  Inventory of Data Sets         EDGRDEXT
  Inventory of VRSs              EDGRKEXT
  Inventory of Owners            EDGROEXT
  Inventory of Products          EDGRPEXT
  Inventory of Racks             EDGRREXT
  Inventory of Stor. Loc. Bins   EDGRSEXT
S Inventory of Scratch Volumes   EDGRVSCR
```

Figure 12-89 Select Report Type panel

```
                        Select Reporting Tool              Row 1 to 2 of 2

S Reporting tool
- ------------------------------
  ICETOOL
  ICETOOL with totals line
```

Figure 12-90 Select Reporting Tool panel

The next step is to select the criteria fields for creating the reports. In the Report Definition panel (Figure 12-91) you define the header, the footer, the reporting tool, and the fields to be reported.

Archived

```
                DFSMSrmm Report Definition - SCRVOL          Row 1 to 17 of 57
Command ===>

Report title  . . . Scratch Volume Inventory
Report footer . . . IBM Internal
Reporting tool  . . ICETOOL                                  Report width:   112

Select the field name to specify a criteria

S CO SO  Field name           Column header text                    CW  Len Typ
- -- --- -------------------- ------------------------------------ --- --- ---
   G  1A RVLCDATE             LAST CHANGE DATE of volume record     33  10   C
   1     RVVOLSER             VOLUME SERIAL NUMBER                  20   6   C
   2     RVLCTIME             LAST CHANGE TIME HHMMSS of record     33   6   C
   3     RVEXPDTO             EXPIRATION DATE - original            27  10   C
   4     RVEXPDT              EXPIRATION DATE - current             26  10   C
   5     RVLCUID              LAST CHANGE USER ID of volume         29   8   C
   6     RVDSNAM1             FIRST FILE DATA SET NAME              24  44   C
      2D RVMDMVID             MULTI-DSET MULTI-VOL ID               22   8   C
         RVCOMP               COMPACTION USED - Y/N                 21   1   C
         RVCRDATE             CREATE DATE of volume record          28  10   C
         RVCRSID              CREATE SYSTEM ID of volume record     33   8   C
         RVCRTME              CREATE TIME HHMMSS of volume record   35   6   C
         RVDEN                RECORDING DENSITY                     17   4   C
         RVDSNNO              NUMBER OF DATASETS ON VOLUME          28   4   C
         RVLCSID              LAST CHANGE SYSTEM ID of volume       31   8   C
         RVNVOL               NEXT VOLUME IN SEQUENCE               23   6   C
         RVPVOL               PREVIOUS VOLUME IN SEQUENCE           27   6   C
         RVTYPE               RECORD TYPE - C'V'                    18   1   C
******************************** Bottom of data ********************************
```

Figure 12-91 Report Definition panel with criteria fields

The previous panel displays a list of the fields in the record and any report criteria specified previously or with the report type. Use the S line command to select the fields for which you would like to view or change the selection criteria. Update the column header text as required for your reporting columns.

You can also enter data in the panel to specify which fields are included in the report, sorted on, and used to group records onto a new page. The grouped fields are used as the primary sort key and to separate the records into pages.

Once you have selected the criteria fields for your report, you see the Report Criteria panel (shown in Figure 12-92) in which you define the criteria for selecting records in your report.

```
                DFSMSrmm Report Criteria - SCRVOL              Row 1 to 3 of 3
Command ===>

Report title: Inventory of Scratch Volumes

The following line commands are valid: B, I, N, P, and T
Comparison operators: EQ =, NE <>, GT >, GE >=, LT <, LE <=, IN, BW
Conjunction: AND, OR, AND(, )AND

S Field name           Op Compare value                          Conj Len Typ
- -------------------- -- ------------------------------------ ---- --- ---
  RVLCDATE             IN 1999/12/31,2000/01/31,2000/02/09,2000/03 AND  10   C
  RVTYPE               EQ V                                     AND   1   C
I RVEXPDT              BW 1999/01/01,2000/12/31                 AND  10   C
******************************** Bottom of data ********************************
```

Figure 12-92 Report Criteria panel

You can use line commands B (Bottom), N (Next), P (Previous), and T (Top) to order the fields in the list.

When you select options **I** in the previous panel, you see the Report Criteria Details panel, showing the details of the field you selected, as shown in Figure 12-93.

```
 DFSMSrmm Report Criteria Details - SCRVOL

Field name . . . : RVEXPDT
Operation  . . . : IN
Compare value(s) : 1999/12/31,2000/01/31,2000/02/09,2000/03/31
Compare value(s) : 2000/04/30
Conjunction  . . : AND
Length . . . . . :  10
Type . . . . . : C
```

Figure 12-93 Report Criteria Details panel

When creating a new report definition by copying an existing one, all the values from the old definition remain the same for the new one.

Working with a reporting JCL

When you select **G** or **J** in the Report Definitions panel, you receive the screen in Figure 12-94 prompting for a data set name containing the records to be reported.

Archived

```
Enter the input data set . . .
```

Figure 12-94 Reporting data set name prompt

If you are generating the JCL, it will be placed in the library named as the user JCL on the options panel.

If you are going to modify a previously created JCL, then you must enter the name of the member or data set name in which the JCL was generated.

Deleting a report definition

When you select **D** in the Report Definitions panel, you receive the screen shown in Figure 12-95 to confirm the delete action.

```
Use ENTER to confirm the Delete, END to cancel.
```

Figure 12-95 Delete report definition confirmation prompt

Displaying or changing a report definition

When you select option **S** in the Report Definitions panel (Figure 12-87 on page 550), you receive the Report Definition panel for the report definition you have selected (Figure 12-91 on page 552) with the details about this definition. In this panel you can change the values you want, and then select new criteria in the Report Criteria panel (Figure 12-92 on page 553).

Working with report types

When you select option **2 (REPORT TYPE)** in the Report Generator panel (Figure 12-85 on page 549), then you receive the Report Types panel shown in Figure 12-96.

Archived

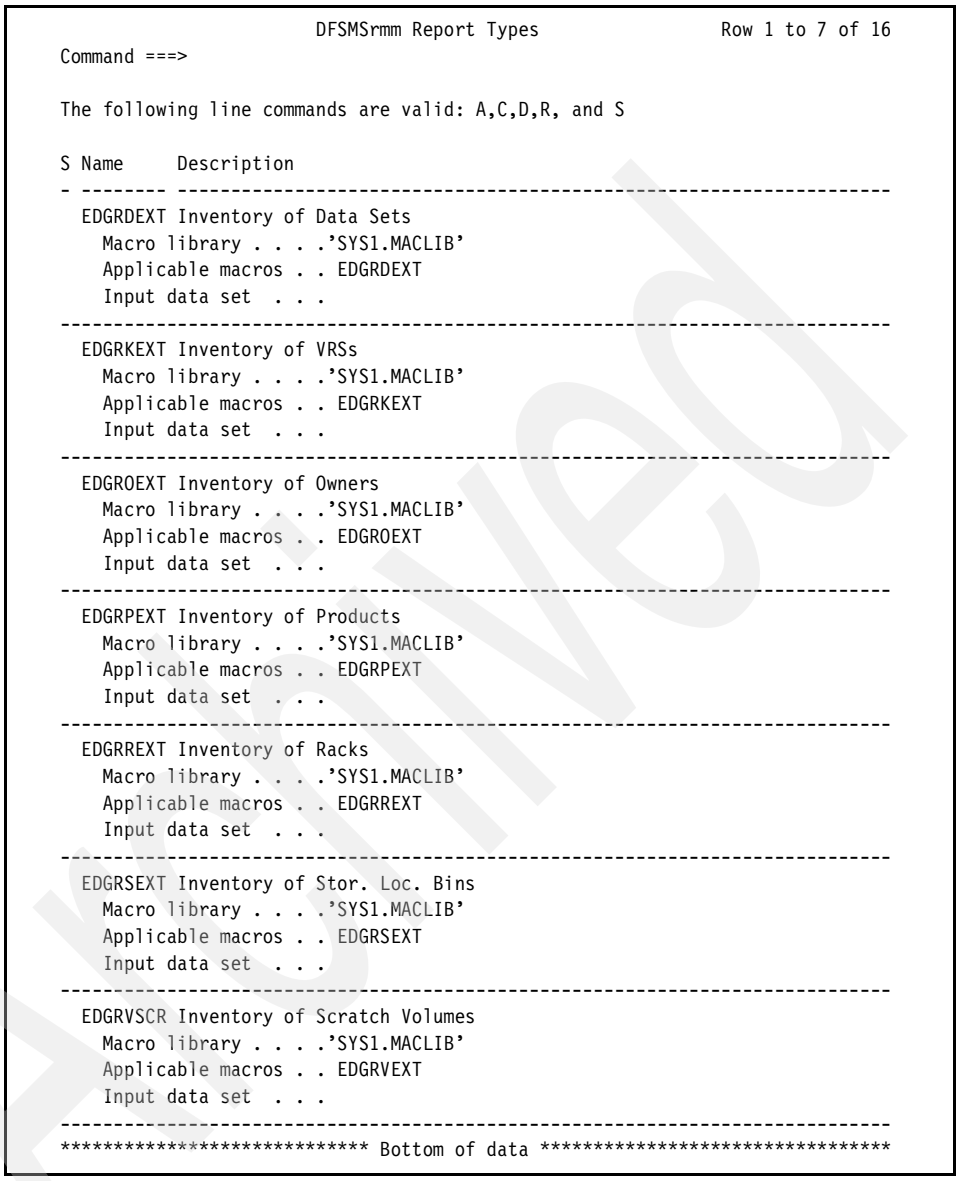


Figure 12-96 Report Types panel

Archived

The available line commands in the previous panel are:

A	Add a new report type
C	Change a report type
D	Delete a report type
R	Add report definition
S	Select criteria

If you select option **S** in the Report Types panel shown in Figure 12-96 on page 555, then you go to the panel shown in Figure 12-97.

```
                DFSMSrmm Report Type - EDGRVSCR              Row 1 to 17 of 57
Command ===>

Report type: Inventory of Scratch Volumes

S Field name           Column header text
- -------------------- -------------------------------------
S RVTYPE               RECORD TYPE - C'V'
S RVSTATUS             VOLUME STATUS
  RVLCDATE             LAST CHANGE DATE of volume record
  RVLCTIME             LAST CHANGE TIME HHMMSS of record
  RVEXPDTO             EXPIRATION DATE - original
  RVEXPDT              EXPIRATION DATE - current
  RVLCUID              LAST CHANGE USER ID of volume
  RVDSNAM1             FIRST FILE DATA SET NAME
  RVPVOL               PREVIOUS VOLUME IN SEQUENCE
  RVNVOL               NEXT VOLUME IN SEQUENCE
  RVMDMVID             MULTI-DSET MULTI-VOL ID
S RVCRDATE             CREATE DATE of volume record
  RVCRTIME             CREATE TIME HHMMSS of volume record
  RVCRSID              CREATE SYSTEM ID of volume record
  RVLCSID              LAST CHANGE SYSTEM ID of volume
  RVDEN                RECORDING DENSITY
  RVCOMP               COMPACTION USED - Y/N
  RVDSNNO              NUMBER OF DATASETS ON VOLUME
******************************** Bottom of data ********************************
```

Figure 12-97 Report Type panel

Once you have selected the fields you want, you go to the Report Type Criteria panel, shown in Figure 12-98, showing the criteria fields you have selected.

Archived

```
                DFSMSrmm Report Type Criteria                  Row 1 to 3 of 3
Command ===>

Report title: Inventory of Scratch Volumes

The following line commands are valid: B, I, N, P, and T
Comparison operators: EQ =, NE <>, GT >, GE >=, LT <, LE <=, IN, and BW
Conjunction: AND, OR, AND(, or )AND

S Field name           Op Compare value                          Conj Len Typ
- -------------------- -- ---------------------------------------- ---- --- ---
  RVTYPE               EQ V                                      AND   1   C
  RVSTATUS             IN SCRATCH,INIT                           AND  10   C
  RVCRDAYE             BW 2000/01/01,2000/12/31                  AND  10   C
******************************** Bottom of data ********************************
```

Figure 12-98 Report Type Criteria panel

The line operators available in this panel are the same as those available in the Report Criteria panel, shown in Figure 12-92 on page 553.

Working with reporting tools

When you select option **3 (REPORTING TOOL)** in the Report Generator panel (Figure 12-85 on page 549), you receive the Reporting Tools panel shown in Figure 12-99.

```
                    DFSMSrmm Reporting Tools                   Row 1 to 2 of 2
Command ===>

The following line commands are valid: A,C, and D

S Reporting tool               Exec     Colspace
- ----------------------------- -------- --------
  ICETOOL                      EDGRGICE        2
  ICETOOL with totals line     ICETOOLT        2
******************************** Bottom of data ***************************
```

Figure 12-99 Reporting Tools panel

Use the Reporting Tools panel to define or change the parameters of a reporting tool. For customizing other than the listed tools, refer to *DFSMSrmm Reporting*.

Archived

12.4.6 Report generator messages

There are new messages introduced with the report generator:

EDGG010	No valid report definition libraries specified
EDGG011	Report definition library does not contain any reports
EDGG012	Report definition library does not exist
EDGG013	User report definition library could not be allocated, RC=rc, RS=rs
EDGG014	Report JCL library could not be allocated, RC=rc, RS=rs
EDGG020	Field name(s) no longer exist in macro(s) - list of field names....
EDGG021	Compare value does not match the field name type
EDGG022	Compare order must be between 1 and 99
EDGG023	Reporting tool name already exists
EDGG024	Report type name already exists
EDGG030	Report definition libraries do not contain any reporting tools
EDGG031	Report definition libraries do not contain any report types
EDGG032	Compare operator allows one compare value only
EDGG033	Compare operator BW requires two compare values
EDGG034	Compare value does not match with the type
EDGG040	Report definition name already exists
EDGG041	Specify the sort order when grouping field names
EDGG042	Sort order not within the range of the number of grouped field names
EDGG043	Macro library not found
EDGG044	Macro name not found in specified library

12.5 Sample reports

There are some sample reporting jobs provided in SYS1.SAMPLIB. They can be very useful for your daily activity. Table 12-3 shows some of these samples, explaining the type of report that is generated.

Archived

Table 12-3 Sample reports

JOB	SAMPLIB member	Report description
RMMCVB	EDGJCVB	Produces RMM CHANGEVOLUME commands for volumes in a store.
RMMAUDIB	EDGJAUDW	Creates a weekly archive from daily audit reports.
RMMAUD2B	EDGJAUDM	Create a monthly archive from weekly audit reports.
RMMRECI1	EDGJRECL	Lists information for identified tapes for a recovery.
RMMRECVL	EDGJRECV	Creates RMM ADDVOLUME commands for identified lost tapes to recover.
RMMSMF	EDGJSMF	Lists all DFSMSrmm SMF volume records in a readable format.
SMFPRTN	EDGJSMFP	Provides a count of each SMF record found.
BARCMDS	EDGJBCAV	Creates RMM ADDVOLUME commands from a list of barcode scanned tapes.
BARCOMP	EDGJCOMB	Create audit reports using barcode scanned output.

For more information about these sample reports and more, refer to *DFSMSrmm Reporting,* SC26-7406.

Archive

Archived

13

Splitting and merging your DFSMSrmm CDS

The control data set contains information about the resources managed by DFSMSrmm. Normally, the resources are all those used on a single system or in a set of systems with a shared DASD environment. If you ever split these systems or merge them together because of changes in workload, you may also want to do the same to the DFSMSrmm control data set.

Objectives of this chapter:

- Understand when to split or merge a DFSMSrmm CDS
- Learn the process to successfully split or merge a DFSMSrmm CDS

At the end of this chapter, you will have an understanding of when to split or merge DFSMSrmm CDSs, and the process to successfully accomplish the task.

© Copyright IBM Corp. 2004. All rights reserved.

13.1 When to merge or split

You should not normally have to merge or split the contents of the DFSMSrmm control data set. However, either during conversion or when you are changing the number of MVS systems you run, you may want to merge or split the contents of the DFSMSrmm control data set.

During conversion

When you are converting from an existing database to an DFSMSrmm control data set, you may want to use an existing DFSMSrmm control data set. You may have one system already converted to DFSMSrmm, and now a second is to be converted, but you want to share the control data set with the first system.

During conversion, you will be adding data produced by EDGCNVT to an existing control data set. You would not be splitting the conversion data stream into multiple control data sets. See 13.1.5, “Converting to an existing CDS” on page 583 for details of how you merge the data you are converting into an existing DFSMSrmm control data set.

Changing the number of systems

If you are changing the number of systems or control data sets that you have, and you are not deleting the information, you will want to merge or split the contents of the DFSMSrmm control data set.

If you are increasing the number of control data sets, you are splitting the control data set. See 13.1.4, “Splitting the CDS” on page 576 for details on how you split an existing control data set into multiple DFSMSrmm control data sets.

If you are decreasing the number of control data sets, you are merging control data sets. See 13.1.3, “Merging the CDS” on page 572 for details on how you merge multiple DFSMSrmm control data sets.

13.1.1 Considerations

When you merge or split a control data set, you must consider the timing of events as well as the contents of the control data set.

DFSMSrmm must be inactive, and you must not be performing tape processing when you are processing the control data set. When you have stopped DFSMSrmm, you should back up existing control data sets and ensure that you have recovery plans in place in case you are unable to complete the process successfully. When you stop DFSMSrmm, do not use the OPT=RESET function, as it allows tapes to be processed without DFSMSrmm’s knowledge.

Archived

When merging or splitting a control data set, it is important to understand the contents of the control data set. For example, if you are merging two control data sets, you cannot merge if the same resource is defined in each control data set. You must analyze the contents of the control data set to ensure that there are no duplicate resources. Below we describe each resource type in the control data set, and explain what you can do to handle duplicate resources.

The record types are:

Key type	Record type
C	Control
CA	Action
D	Data set
E	Empty rack
F	Scratch rack
K	VRS
O	Owner
P	Product
R	Empty bin
S	In-use bin
U	In-use rack
V	Volume

Control record

Each DFSMSrmm control data set contains a control record. The control record contains the dates and times of when inventory management functions last ran, and the counts of rack numbers and bin numbers for built-in storage locations.

You can merge the control record from a single control data set and copy it to all target control data sets if you are splitting a control data set. In both cases you must use EDGUTIL with PARM=MEND to correct the record counts after processing is completed.

The control record also contains the CDSID value. If you use this value to identify a control data set, you must ensure that each control data set you create uses a different value. Use EDGUTIL with PARM='UPDATE' after processing is completed and specify the CDSID value, using the `SYSIN DD CONTROL` statement.

Action record

Action records contain information about the librarian actions that are outstanding. They include the release actions and any volume movements that are required. Omit these records, because there is a strong chance that they exist in each control data set, and the records are rebuilt by DFSMSrmm during inventory management.

Archived

Data set record

A data set record exists for each volume on which the data set has been written. Each record is uniquely identified by the VOLSER and physical file number of the data set.

When merging control data sets, refer to the information on volume records in “Volume record” on page 569 for an explanation of how to handle duplicate volumes and the data sets on them.

When splitting a control data set, you must copy all data set records to each target control data set, and then use the facilities of DFSMSrmm to delete those data set records that are not required. The DFSMSrmm EDGUTIL utility determines which are required by using the volume information you have copied into each target control data set. Use the PARM=MEND option to delete the unnecessary data set records from the control data set.

VRS record

The VRS records contain the retention and movement policies for your data. If you have multiple control data sets, it is likely that you have the same policies defined in each control data set for some types of data. The policies are identified by data set name and job name, VOLSER, and the VOLSER prefix.

When you merge control data sets, you must check for and resolve any duplicate policies. If you have a policy for the same data set name and job name combination in more than one control data set, you should check that the retention and movement defined are the same. If they are the same, you can use the policy from any control data set. If they are not the same, you must resolve the conflict before merging the control data sets.

One way to resolve the conflict would be to ensure that one control data set contains the complete set of VRS policies. Identify the control data set with most of the VRSs you require in the merged control data set. Identify the VRSs that are unique in the other control data sets, and define them in the one control data set.

When you split a control data set, you copy the complete set of VRS records to all target control data sets. This action may result in having VRSs that are unused in some control data sets, but it does not cause a problem in DFSMSrmm processing. Once DFSMSrmm is started using the new control data set, you can use the **DELETEVRS** subcommand to delete the unused VRSs.

Owner record

There is one owner record for each identified user that owns a DFSMSrmm resource such as a volume or VRS. There can also be owner records for users

that own no resources. Perhaps you have defined these so that the information can be used within your installation.

The owner record contains detailed information about the user and the count of volumes owned. For users who own volumes, DFSMSrmm uses multiple owner records to provide directed retrieval of volume and data set information during search commands. You must consider the owner details and the owner volume records when splitting and merging the control data set.

Whether you are merging or splitting control data sets, you must use a SORT program to select the owner records, remove the additional owner records that contain just volume information, and remove the duplicate owner records. Once the records are loaded into a control data set, use the EDGUTIL utility with the PARM=MEND option to set the correct counts of owned volumes and build the owner volume records for owned volumes.

When merging control data sets, you must check for duplicate owners across the control data sets being merged. Duplicates are not a problem, as they can be handled as long as the owner record identifies the same person or task in each control data set. If the duplicate owners are the same user, determine which control data set has the correct owner details, such as address and contact information. Ensure that at least one control data set has the correct owner details for the users, and use the owner details as the primary source of input to the merge. If the duplicate owners are really different users, you must decide how to handle that situation. You may have to assign different user IDs or define a new owner ID and transfer the owned resources to the new owner ID. Your objective is to have no duplicate owners, or to have any duplicates be for the same user in each case.

Product record

A product record identifies a single product and software level, and the volumes associated with that product. Whether you are merging or splitting a control data set, the existence of product records is a problem that must be resolved.

When you merge control data sets, there must be no duplicate product records. If there are, you must delete the duplicates and redefine the product information after the merge is completed. For each duplicate product that must be deleted, list the volumes for that product. Example 13-1 shows how to list the volumes, assuming that all product information is being deleted from a single control data set.

Example 13-1 Listing product information

```
RMM SEARCHPRODUCT NUMBER(*) LIMIT(*) NOLIST -
    CLIST('RMM LISTPRODUCT ')
EXEC EXEC.RMM LIST
```

Example 13-2 shows how to delete all product information from a single control data set.

Example 13-2 Deleting product information

```
RMM SEARCHPRODUCT NUMBER(*) LIMIT(*) NOLIST -
    CLIST('RMM DELETEPRODUCT ',' NORELEASE')
EXEC EXEC.RMM LIST
```

After you merge the control data sets, you can add the product information using the RMM **CHANGEVOLUME** subcommand as shown in Example 13-3.

Example 13-3 Adding product information

```
RMM CHANGEVOLUME DZ11CO NUMBER(5695-DF1) FEATCD(A001) -
    LEVEL(V01R03M00)
```

When you split a control data set, the easiest way to handle product records is to target all product volumes to a single control data set. If this is not possible, you can either:

- Delete the product information and then add it after the control data set is split.
- Individually copy the product records by key to the correct target control data set. You can do this only if all volumes for a product are targeted to the same control data set.

Rack numbers record

There is a single rack record for each shelf location you have defined to DFSMSrmm. There must be at least one for each volume defined, but there can be more if you also defined empty rack numbers for use when volumes are added to the control data set. The rack number represents the volume's external VOLSER.

There are three rack number records, each using a unique key to aid retrieval by DFSMSrmm. During the copy operation, the three key ranges must be copied, but you only need to consider the rack numbers that you have defined.

When you merge control data sets, you must not have any duplicate rack numbers. Any duplicates must be deleted. If the duplicate already contains a

Archived

volume, you can reassign the volume to another rack number by using the RMM **CHANGEVOLUME** subcommand. Example 13-4 shows two different ways of reassigning a volume to a new rack number.

Example 13-4 Assigning a new rack number

```
RMM CHANGEVOLUME PP1234 POOL(A*)

RMM CHANGEVOLUME PP1234 RACK(A00001)
```

In the first command, DFSMSrmm picks an empty rack in pool A*. In the second example, you have selected the empty rack to be used.

Before cleaning up any duplicate rack numbers, refer to “Volume record” on page 569 for an explanation of how to handle duplicate volumes.

When you are splitting a control data set, you split the rack numbers in the same way that you split the volume ranges into the target control data sets. You decide which volumes are copied to each control data set, and ensure that the rack numbers with which they are associated are also copied to the same target control data set. During the copy process you identify the rack numbers using key ranges to include or exclude the appropriate rack numbers. The significant part of the key of rack number records is the first 16 characters. The key must be specified in hexadecimal, as the second byte is hex zero. When copying ranges of volumes, you can use the COUNT keyword on the REPRO statement; however, you must specify a hexadecimal key for the TOKEY as well as the FROMKEY. The key includes an 8-character media name that matches the VLPOOL media name as well as the rack number. Example 13-5 shows an example of copying the rack numbers for the volumes in the 001000 - 001999 range.

Example 13-5 Copying rack numbers

```
REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'C500F3F4F8F040404040F0F0F1F0') -
        TOKEY(X'C500F3F4F8F040404040F0F0F1F9F9F9')
 /*  COPY racks from E 3480    001000       to  E  3480    001999 */
 REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'C600F3F4F8F040404040F0F0F1F0') -
        TOKEY(X'C600F3F4F8F040404040F0F0F1F9F9F9')
 /*  COPY racks from F 3480    001000       to  F  3480    001999 */
 REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'E400F3F4F8F040404040F0F0F1F0') -
        TOKEY(X'E400F3F4F8F040404040F0F0F1F9F9F9')
 /*  COPY racks from U 3480    001000       to  U  3480    001999 */
```

The first character of the key is either E, F, or U, the media name starts in position 3, and the rack number starts in position 11. Only copy a range of volumes to a single target control data set.

Bin numbers record

There is a single bin record for each storage location shelf location you have defined to DFSMSrmm.

There are two different bin number records, each using a unique key to aid retrieval by DFSMSrmm. During the copy operation, the two key ranges must be copied, but you only need to consider the bin numbers that you have defined.

When you merge control data sets, you must not have any duplicate bin numbers. Any duplicates must be deleted. If the duplicate already contains a volume, you can reassign the volume to another bin number by using the RMM **CHANGEVOLUME** subcommand. You can reassign a volume to a new bin number issuing the command shown in Example 13-6.

Example 13-6 Change the assigned bin number

```
RMM CHANGEVOLUME A00123 BIN(000029)
```

You can also remove the volume from the storage location, free up the bin number, and delete the unused bin number. Example 13-7 shows how to delete duplicate bin number X00022 in location VAULT1 of media name CARTS.

Example 13-7 Freeing and deleting a duplicate bin number

```
RMM CHANGEVOLUME A00123 LOCATION(HOME) CMOVE
RMM DELETEBIN X00022 MEDIANAME(CARTS) LOCATION(VAULT1)
```

The next time you run inventory management, the volume is reassigned to the storage location, and another bin number is assigned.

Note: When you need to reassign bin numbers, you should consider how offsite movements are managed in your installation. If you depend on the volume being in a specific slot in a storage location, you should ensure that any volumes you change are physically moved in the storage location.

When you are splitting a control data set, you split the bin numbers according to how you want to handle storage locations in the split environment. If the volumes in the storage locations are being split such that each target control data set contains volumes for each shelf-managed storage location, handling the splitting of bin numbers is very complex. We suggest that you use DFSMSrmm commands to change the location of all volumes and confirm their return to their

home location. This approach frees up the in-use bin numbers. Do not copy the bin numbers to the target control data sets, but add empty bin numbers after the split and use inventory management VRSEL and DSTORE processing to assign new bin numbers to the volumes after the split processing is completed.

Targeting all volumes in a single storage location for the same control data set simplifies the copying of bin numbers. During the copy process, you identify the bin numbers using key ranges to include or exclude the appropriate bin numbers. The significant part of the key of bin number records is the first 24 characters. The key can be specified in character form. When copying ranges of bin numbers, you must specify a TOKEY as well as the FROMKEY because a range of bin numbers can include both empty and in-use bin numbers. The key includes an 8-character media name, which matches the LOCDEF media name as well as the bin number, and the location name. Figure 13-8 shows an example of copying the bin numbers for the bin numbers in storage location VLT1, media name CARTS, and the V10001 - V10500 range.

Example 13-8 Sample REPRO statements for copying bin numbers

```
REPRO INFILE(SOURCE) OUTFILE(OUTS1)        -
      FROMKEY(C'RUVLT1    CARTS   V10001') -
        TOKEY(C'RUVLT1    CARTS   V10500')
REPRO INFILE(SOURCE) OUTFILE(OUTS1)        -
      FROMKEY(C'SUVLT1    CARTS   V10001') -
        TOKEY(C'SUVLT1    CARTS   V10500')
```

The first two characters of the key are either RU or SU, the storage location is in position 3, the media name starts in position 11, and the bin number starts in position 19. Only copy a range of bin numbers to a single target control data set.

Volume record

The volume record contains detailed information about a volume and its contents. It also identifies the data sets that are on the volume and volume chaining information. Some fields in the records identify the existence of other records in the control data set.

When moving volume records from one control data set to another, you must be sure to move the related records as well. These include the data set records, owner record, rack record, bin records, and product record. If you do not move them, the EDGUTIL utility can identify the missing records, and with the MEND option, creates the missing records with one exception which is the product records that cannot be re-created.

When you are merging control data sets, you should not have any duplicate volumes. If you do, you must either skip the duplicate volumes or delete them from the source control data set before starting the merge process. You should

understand why there are duplicate volumes, and based on your understanding, select the best approach for dealing with them.

If you have duplicate volumes, there can also be duplicate data set records for those volumes. If you delete the volumes, you also delete the data set records.

When you are splitting a control data set, you will most probably split the volumes according to VOLSER ranges. During the copy process you identify the volumes using key ranges to include or exclude the appropriate volumes. The significant part of the key of volume records is the first eight characters. The key must be specified in hexadecimal as the second byte is hex zero. When copying ranges of volumes, you can use the COUNT keyword on the REPRO statement, otherwise, you must specify a hexadecimal key for the TOKEY as well. Example 13-9 shows an example of copying the volumes in the 001000 - 001999 range.

Example 13-9 Sample REPRO statements for copying ranges of volumes

```
REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'E500F0F0F1') COUNT(1000)
```

The first character of the key is V, and the VOLSER starts in position 3. Only copy a range of volumes to a single target control data set.

13.1.2 PARMLIB options

The PARMLIB options that are affected by the splitting and merging of control data sets are:

- LOCDEF
- VLPOOL
- REJECT

LOCDEF

These options identify your locations to DFSMSrmm. When you are merging control data sets, consolidate the LOCDEF options from all control data sets you are merging to ensure that you have no duplicates. If you have duplicates, ensure that they are identified as the same location type, and with the same media names.

When you split a control data set, each system that will use one of the target control data sets should have defined to it the locations where the volumes in the target control data set reside. You can simply include all LOCDEFs in each new system, but you have the option of removing those that are not required.

VLPOOL

These options identify the ranges of rack numbers and pools of shelf space that you have in your library. If your library is not changing, we recommend that you do not change your current VLPOOL options.

If you are merging control data sets and your existing VLPOOL definitions are different for each existing control data set, you must merge the definitions together to cover the merged sets of volumes and rack numbers. Ensure that the operands specified on each VLPOOL are correct when the same pool prefix is defined in more than one source environment.

When you split a control data set, we recommend that you take the existing VLPOOL definitions and use them on each of the new control data sets.

REJECT

These options identify the ranges of volumes to be rejected for use on the system. The reject specifies the rack number prefix and should match the VLPOOL definitions that you have.

Where you have a tape library shared by multiple systems, define the same VLPOOL definitions to all systems, and use the REJECT definitions to prevent systems from using volumes that are for use on other systems.

The actions you take for the REJECT definitions should match how you handle the VLPOOL definitions.

Running IDCAMS REPRO

When using IDCAMS REPRO to copy records to the target control data set, you may get messages indicating that records are duplicates, or that there are no input records:

- Return code 4 from a **REPRO** command with this message:
  IDC11465I DELIMITERS WERE SPECIFIED, BUT NO RECORDS PROCESSED.

  The only key range for which this should occur is *P* for product records. Expect this message if you do not have any product definitions in the control data set.
- Return code 8 or 12 from a **REPRO** command with messages:
  IDC3302I ACTION ERROR ON DFRMM.MASTER.MERGE
  IDC3308I ** DUPLICATE RECORD X'ccXXXXXXXX'

  You should not get these messages if you have followed the recommendations made above. After four duplicate records, IDCAMS stops processing the records, so the copy operation is incomplete. You must resolve

Archive

the duplicates by taking the actions recommended in 13.1.1, “Considerations” on page 562 and then rerun the merge or split job.

13.1.3 Merging the CDS

Follow the steps outlined below to perform the merge of two control data sets. SYS1 is the system on which the merge is being performed, and SYS2 is the second system from which the control data set data is being merged.

1. Refer to 13.1.1, “Considerations” on page 562 and use the information to analyze the control data sets that are being merged. If any potential problems or duplicate records are identified, you must make the recommended changes before continuing.
2. Stop DFSMSrmm on all systems affected by the merging of the control data sets. You cannot use tapes while you merge control data sets.
3. Back up the control data set on SYS2 and send the backup to SYS1. Run the backup job shown in Example 13-10. If you have shared DASD between SYS1 and SYS2, you can skip this step.
4. Restore the SYS2 control data set on SYS1. Run the restore job shown in Example 13-11. If you have shared DASD between SYS1 and SYS2, you can skip this step.
5. Run the merge job shown in Example 13-12. The job includes the following steps:

BACKS1	Backs up the control data set used for SYS1
COPYO	Copies the owner records from both source control data sets to temporary files for input to step SORTO
SORTO	Removes any duplicate owner records found in SYS2 input and deletes the owner volume records
DEFINE	Deletes and then redefines a target control data set
COPY	Copies most of the records from both source control data sets. For VRS and product records, only one control data set is used as input, and a later step copies from the second control data set. This is done as these are the most likely records to still have duplicates.
COPYK2	Copies the VRS records from SYS2. Expect duplicates to be found if you use the same VRS definitions on both control data sets. You can remove this step if you have already ensured that all VRS definitions were in SYS1.
COPYP2	Copies the product records from SYS2.
MEND	Validates the newly merged control data set and makes any corrections to record relationships. Builds the owner volume records, corrects the count of owned volumes, and corrects the rack number and bin number counts in the control record.

VERIFY Verifies that the control data set is logically sound and ready to use.

If the merge job is successful, you are ready to use the merged control data set.

6. Update SYS1.PARMLIB member EDGRMMxx:
   Command Required change

   OPTION Update DSNAME to identify the new control data set name.
   LOCDEF Update to reflect all locations used in the merged control data set.
   VLPOOL Update to reflect the new set of pools.
   REJECT Update to reflect any changes in VLPOOL definitions.
7. Start the DFSMSrmm procedure using the new PARMLIB member.
8. Check that the merged information is good by using DFSMSrmm commands, and the ISPF dialog to retrieve and display information.
9. Execute any commands that are identified to be run in 13.1, “When to merge or split” on page 562.
10. Run DFSMSrmm inventory management on the new control data set. During this run, DFSMSrmm VRSEL and DSTORE processing identifies volumes to be moved to the correct location. If any moves are identified, use EDGRPTD to produce the picking lists and actions of any moves that you know are still required.

    The inventory management run should execute all inventory management functions including backup.

Example 13-10 shows the JCL for backing up the control data set on SYS2.

Example 13-10 JCL to back up the control data set on SYS2

```
//BACKS2    EXEC PGM=EDGBKUP,PARM='BACKUP'
//SYSPRINT  DD   SYSOUT=*
//SYSPRINT  DD   SYSOUT=*
//MASTER    DD   DSN=DFRMM.MASTER.SYS2,DISP=SHR
//BACKUP    DD   DSN=DFRMM.MASTER.BACKUP2(+1),DISP=(,CATLG,DELETE),
//         UNIT=SYSDA,SPACE=(6233,(1000,999),RLSE),
//         DCB=(MODEL.VB9216,DSORG=PS)
```

Example 13-13 shows the JCL for restoring the SYS2 control data set on SYS1.

Example 13-11 JCL to Restore the SYS2 control data set on SYS

```
//RESTS2    EXEC PGM=EDGBKUP,PARM='RESTORE'
//SYSPRINT  DD   SYSOUT=*
//MASTER    DD   DSN=DFRMM.MASTER.SYS2,DISP=SHR
```

Archived

```
//BACKUP    DD   DSN=DFRMM.MASTER.BACKUP2(0),DISP=SHR
```

Example 13-12 shows the JCL for executing the merge process for two control data sets.

Example 13-12 Merge JCL

```
//BACKS1   EXEC PGM=EDGBKUP,PARM='BACKUP'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   DSN=DFRMM.MASTER.SYS1,DISP=SHR
//BACKUP   DD   DSN=DFRMM.MASTER.BACKUP1(+1),DISP=(,CATLG,DELETE),
//         UNIT=SYSDA,SPACE=(6233,(1000,999),RLSE),
//         DCB=(MODEL.VB9216,DSORG=PS)
//*************************************************************
//** COPY THE OWNER RECORDS FROM BOTH SOURCE CDSs TO          **
//** A SEQUENTIAL DATASET TO BE USED FOR THE MERGED CDS.      **
//*************************************************************
//COPY0    EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//INS1     DD   DSN=DFRMM.MASTER.SYS1,DISP=SHR
//INS2     DD   DSN=DFRMM.MASTER.SYS2,DISP=SHR
//OUTS1    DD   DSN=&&IN01,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=(1024,(5000,5000),RLSE),
//         LRECL=2048,RECFM=VB
//OUTS2    DD   DSN=&&IN02,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=(1024,(5000,5000),RLSE),
//         LRECL=2048,RECFM=VB
//SYSIN    DD   *
 REPRO INFILE(INS1) OUTFILE(OUTS1) FROMKEY(0) TOKEY(0)
 REPRO INFILE(INS2) OUTFILE(OUTS2) FROMKEY(0) TOKEY(0)
//*************************************************************
//** SORT THE OWNER  RECORDS FROM BOTH SYSTEMS TO A TEMPORARY  **
//** SEQUENTIAL DATASET TO BE PASSED TO A LATER STEP COPYALL.  **
//** REMOVE DUPLICATE OWNERS FROM SOURCE S2.                  **
//*************************************************************
//SORT0    EXEC PGM=SORT,REGION=6M
//SYSPRINT DD   SYSOUT=*
//SORTIN   DD   DSN=&&IN01,DISP=(OLD,DELETE)
//         DD   DSN=&&IN02,DISP=(OLD,DELETE)
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTOUT  DD   DISP=(NEW,PASS),UNIT=SYSDA,DCB=*.SORTIN,
//         SPACE=(1024,(5000,5000),RLSE),DSN=&&SORT0
//SYSOUT   DD   SYSOUT=*
//SYSIN    DD   *
 SORT FIELDS=(5,9,CH,A)
 INCLUDE COND=(5,1,CH,EQ,C'O',AND,14,1,CH,EQ,X'00')   BASE OWNERS ONLY
 SUM  FIELDS=NONE                                       ELIMINATE DUPS
//DEFINE   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
```

```
//MSTVOL   DD  DISP=SHR,UNIT=3380,VOL=SER=VSMN01
//SYSIN    DD  *
  DEL DFRMM.MASTER.MERGE
  SET MAXCC=0
  DEF CL(NAME(DFRMM.MASTER.MERGE) FILE(MSTVOL)             -
         VOLUME(VSMN01) KILOBYTES(50000 9000) FSPC(20 20) -
         RECORDSIZE(512 9216)  KEYS(56 0) SHR(3 3) RUS)  -
       DATA(NAME(DFRMM.MASTER.MERGE.DATA)           -
            CISZ(10240) )                 -
       INDEX(NAME(DFRMM.MASTER.MERGE.INDEX) IMBED REPL     -
            CISZ(2048) )
//*************************************************************
//**              TARGET SYSTEM                              **
//** COPY THE RECORDS FROM BOTH SOURCE CDS TO THE TARGET CDS    **
//*************************************************************
//COPY     EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//INS1     DD   DSN=DFRMM.MASTER.SYS1,DISP=SHR
//INS2     DD   DSN=DFRMM.MASTER.SYS2,DISP=SHR
//TARGET   DD   DSN=DFRMM.MASTER.MERGE,DISP=SHR
//OWNER    DD   DSN=&&SORTO,DISP=(OLD,DELETE)
//SYSIN    DD   *
 REPRO INFILE(INS1) OUTFILE(TARGET) FROMKEY(C) TOKEY(C) COUNT(1)
 REPRO INFILE(INS1) OUTFILE(TARGET) FROMKEY(D) TOKEY(F)
 REPRO INFILE(INS2) OUTFILE(TARGET) FROMKEY(D) TOKEY(F)
 REPRO INFILE(INS1) OUTFILE(TARGET) FROMKEY(K) TOKEY(K)
 REPRO INFILE(OWNER) OUTFILE(TARGET)
 REPRO INFILE(INS1) OUTFILE(TARGET) FROMKEY(P) TOKEY(P)
 REPRO INFILE(INS1) OUTFILE(TARGET) FROMKEY(R) TOKEY(V)
 REPRO INFILE(INS2) OUTFILE(TARGET) FROMKEY(R) TOKEY(V)
//*************************************************************
//**              SOURCE CDS 2 - ON TARGET SYSTEM             **
//** COPY "K" RECORDS FROM THE SOURCE CDS 2 TO THE TARGET CDS.  **
//** RETURN CODE 8 or 12 INDICATES DUPLICATE RECORDS.           **
//*************************************************************
//COPYK2   EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//INS2     DD   DSN=DFRMM.MASTER.SYS2,DISP=SHR
//TARGET   DD   DSN=DFRMM.MASTER.MERGE,DISP=SHR
//SYSIN    DD   *
 REPRO INFILE(INS2) OUTFILE(TARGET) FROMKEY(K) TOKEY(K)
//*************************************************************
//**              SOURCE CDS 2 - ON TARGET SYSTEM             **
//** COPY "P" RECORDS FROM THE SOURCE CDS 2 TO THE TARGET CDS.  **
//** RETURN CODE 8 or 12 INDICATES DUPLICATE RECORDS.           **
//*************************************************************
//COPYP2   EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//INS2     DD   DSN=DFRMM.MASTER.SYS2,DISP=SHR
```

```
//TARGET   DD   DSN=DFRMM.MASTER.MERGE,DISP=SHR
//SYSIN    DD   *
 REPRO INFILE(INS2) OUTFILE(TARGET) FROMKEY(P) TOKEY(P)
//*************************************************************
//**              MERGED TARGET CDS                          **
//** RUN EDGUTIL WITH   PARM='MEND' TO UPDATE THE NEW MERGED  **
//**   CDS TO HAVE THE CORRECT COUNTS FOR RACK/BINS.          **
//**   ALSO SETS OWNER VOLUME COUNTS AND BUILDS OWNER VOLUME  **
//**   INFORMATION.                                           **
//*************************************************************
//MEND     EXEC PGM=EDGUTIL,PARM='MEND'
//MASTER   DD   DISP=SHR,DSN=DFRMM.MASTER.MERGE
//SYSPRINT DD   SYSOUT=*
//*************************************************************
//**              MERGED TARGET CDS                          **
//** RUN EDGUTIL WITH   PARM='VERIFY(ALL)' ON THE NEW MERGED  **
//**   CDS TO VERIFY THE INTEGRITY OF THE NEW CDS.            **
//*************************************************************
//VERIFY   EXEC PGM=EDGUTIL,PARM='VERIFY(ALL)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   SUBSYS=(BLSR,'DDNAME=MASTERB')
//MASTERB  DD   DISP=SHR,DSN=DFRMM.MASTER.MERGE
//* Use of Batch LSR can speed the running of EDGUTIL
//* IF you have not got BLSR setup on your system you must replace
//* the MASTER DD statement with the following one, and delete
//* the MASTERB DD statement
//*MASTER  DD  DISP=OLD,DSN=DFRMM.MASTER.MERGE,
//*            AMP=('BUFND=50,BUFNI=50')
//SYSIN    DD  DUMMY
```

13.1.4 Splitting the CDS

Follow the steps outlined below to split a control data set into two control data sets. SYS1 and SYS2 are the new systems to which control data set data is being split:

1. Refer to 13.1.1, “Considerations” on page 562 and use the information to analyze the control data set that is being split. If any potential problems are identified, you must make the recommended changes before continuing. You need to consider how you plan to split the ranges of volumes, rack numbers, and bin numbers. You must prepare the AMS statements for use in job steps COPYV and COPYB.
2. Stop DFSMSrmm on all systems affected by the splitting of the control data set. You cannot use tapes while you split a control data set.
3. Run the merge job shown in Example 13-13. The job includes the following steps:

BACKS	Backs up the control data set to be split
COPYO	Copies the owner records from the source control data set to a temporary file for input to step SORTO
SORTO	Removes the owner volume records
DEFINE	Deletes and then redefines the target control data sets
COPY	Copies those records from the source control data set that must be copied to both target control data sets. The copying of the record ranges that need to be split are performed in later steps
COPYP2	Copies the product records to SYS2. This assumes that all product volumes are split out to SYS2 control data set. If this is not the case, refer to “Product record” on page 565 and update the JCL based on the recommended actions.
COPYV	Copies the volume and rack numbers to the two target control data sets. The JCL example shows how you can specify to AMS which records are to be copied. You must update the sample AMS statements to copy the actual ranges you want to have in each target control data set.
COPYB	Copies the bin numbers to the two target control data sets. The JCL example shows how you can specify to AMS which records are to be copied. You must update the sample AMS statements to copy the actual ranges you want to have in each target control data set.
MEND1	Validates the newly created SYS1 control data set and makes any corrections to record relationships. Builds the owner volume records, corrects the count of owned volumes, corrects the rack number and bin number counts in the control record, and deletes any unused data set records.
MEND2	Validates the newly created SYS2 control data set and makes any corrections to record relationships. Builds the owner volume records, corrects the count of owned volumes, corrects the rack number and bin number counts in the control record, and deletes any unused data set records.
VERIFY1	Verifies that the SYS1 control data set is logically sound and ready to use.
VERIFY2	Verifies that the SYS2 control data set is logically sound and ready to use.

If the merge job is successful, you are ready to use the new control data sets.

4. Back up the newly created control data sets for SYS1 and SYS2. Run the backup job shown in Example 13-14.

5. Update SYS1.PARMLIB member EDGRMMxx on all affected systems:

Command	Required change
OPTION	Update DSNAME to identify the new control data set name.

Archive

OPTION Update SYSID to identify a unique system ID. If you do not specify a value, DFSMSrmm uses the SMF system ID.

OPTION If you specified CDSID in PARMLIB for the source control data set, ensure that CDSID still matches the value you used on the source control data set.

LOCDEF Update to reflect the locations used in each control data set.

VLPOOL Update to reflect the new set of pools. Update the SYSID to match the value specified on the OPTION SYSID keyword. If SYSID is not specified, ensure that the VLPOOL SYSID, if required, matches the SMF system ID of the running system.

REJECT Update to reflect any changes in VLPOOL definitions.

6. Start the DFSMSrmm procedure on each system, using the new PARMLIB member.
7. Check that the merged information is good by using DFSMSrmm commands and the ISPF dialog to retrieve and display information.
8. Execute any commands that are identified to be run in 13.1, “When to merge or split” on page 562.
9. Run DFSMSrmm inventory management on each of the new control data sets. During this run DFSMSrmm, VRSEL and DSTORE processing identifies volumes to be moved to the correct location. If any moves are identified, use EDGRPTD to produce the picking lists and actions of any moves that you know are still required.

   The inventory management run should execute all inventory management functions including backup.

Example 13-13 shows the JCL for executing the split process to create two target control data sets.

Example 13-13 Sample JCL to split a DFSMSrmm CDS

```
//BACKS     EXEC PGM=EDGBKUP,PARM='BACKUP'
//SYSPRINT  DD   SYSOUT=*
//MASTER    DD   DSN=DFRMM.MASTER.CDS,DISP=SHR
//BACKUP    DD   DSN=DFRMM.MASTER.BACKUP(+1),DISP=(,CATLG,DELETE),
//         UNIT=SYSDA,SPACE=(6233,(1000,999),RLSE),
//         DCB=(MODEL.VB9216,DSORG=PS)
//*************************************************************
//** COPY THE OWNER RECORDS FROM SOURCE CDS TO                **
//** A SEQUENTIAL DATASET TO BE USED FOR THE NEW CDSs         **
//*************************************************************
//COPYO    EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//SOURCE   DD   DSN=DFRMM.MASTER.CDS,DISP=SHR
//OUT      DD   DSN=&&OUTO,DISP=(,PASS),UNIT=SYSDA,
//         SPACE=(1024,(5000,5000),RLSE),
```

```
//         LRECL=2048,RECFM=VB
//SYSIN    DD    *
 REPRO INFILE(SOURCE) OUTFILE(OUT) FROMKEY(0) TOKEY(0)
//*************************************************************
//** SORT THE OWNER  RECORDS FROM BOTH SYSTEMS TO A TEMPORARY   **
//** SEQUENTIAL DATASET TO BE PASSED TO A LATER STEP COPYALL.   **
//** REMOVE DUPLICATE OWNERS FROM SOURCE S2.                    **
//*************************************************************
//SORTO    EXEC PGM=SORT,REGION=6M
//SYSPRINT DD   SYSOUT=*
//SORTIN   DD   DSN=&&OUTO,DISP=(OLD,DELETE)
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTOUT  DD   DISP=(NEW,PASS),UNIT=SYSDA,DCB=*.SORTIN,
//         SPACE=(1024,(5000,5000),RLSE),DSN=&&SORTO
//SYSOUT   DD   SYSOUT=*
//SYSIN    DD   *
 SORT FIELDS=(5,9,CH,A)
 INCLUDE COND=(5,1,CH,EQ,C'O',AND,14,1,CH,EQ,X'00')   BASE OWNERS ONLY
 SUM  FIELDS=NONE                                     ELIMINATE DUPS
//DEFINE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYS1VOL  DD   DISP=SHR,UNIT=3380,VOL=SER=VSMN01
//SYS2VOL  DD   DISP=SHR,UNIT=3380,VOL=SER=VSMN02
//SYSIN    DD   *
  DEL DFRMM.NEW.SYS1
  DEL DFRMM.NEW.SYS2
  SET MAXCC=0
  DEF CL(NAME(DFRMM.NEW.SYS1) FILE(SYS1VOL)              -
         VOLUME(VSMN01) KILOBYTES(50000 9000) FSPC(20 20) -
         RECORDSIZE(512 9216)  KEYS(56 0) SHR(3 3) RUS)   -
       DATA(NAME(DFRMM.NEW.SYS1.DATA)                    -
            CISZ(10240) )                                -
       INDEX(NAME(DFRMM.NEW.SYS1.INDEX) IMBED REPL       -
            CISZ(2048) )
  DEF CL(NAME(DFRMM.NEW.SYS2) FILE(SYS2VOL)              -
         VOLUME(VSMN02) KILOBYTES(50000 9000) FSPC(20 20) -
         RECORDSIZE(512 9216)  KEYS(56 0) SHR(3 3) RUS)   -
       DATA(NAME(DFRMM.NEW.SYS2.DATA)                    -
            CISZ(10240) )                                -
       INDEX(NAME(DFRMM.NEW.SYS2.INDEX) IMBED REPL       -
            CISZ(2048) )
//*************************************************************
//**              SOURCE SYSTEM                               **
//** COPY THE RECORDS FROM SOURCE TO BOTH TARGET CDSs          **
//*************************************************************
//COPY     EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//OUTS1    DD   DSN=DFRMM.NEW.SYS1,DISP=SHR
//OUTS2    DD   DSN=DFRMM.NEW.SYS2,DISP=SHR
```

Archived

```
//SOURCE   DD   DSN=DFRMM.MASTER.CDS,DISP=SHR
//OWNER    DD   DSN=&&SORTO,DISP=(OLD,DELETE)
//SYSIN    DD   *
 REPRO OUTFILE(OUTS1) INFILE(SOURCE) FROMKEY(C) TOKEY(C) COUNT(1)
 REPRO OUTFILE(OUTS2) INFILE(SOURCE) FROMKEY(C) TOKEY(C) COUNT(1)
 REPRO OUTFILE(OUTS1) INFILE(SOURCE) FROMKEY(D) TOKEY(D)
 REPRO OUTFILE(OUTS2) INFILE(SOURCE) FROMKEY(D) TOKEY(D)
 REPRO OUTFILE(OUTS1) INFILE(SOURCE) FROMKEY(K) TOKEY(K)
 REPRO OUTFILE(OUTS2) INFILE(SOURCE) FROMKEY(K) TOKEY(K)
 REPRO INFILE(OWNER) OUTFILE(OUTS1)
 REPRO INFILE(OWNER) OUTFILE(OUTS2)
//*************************************************************
//**             SOURCE TO A SINGLE TARGET                   **
//** COPY "P" RECORDS FROM THE SOURCE CDS TO THE ONE TARGET    **
//*************************************************************
//COPYP2   EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//OUTS2    DD   DSN=DFRMM.NEW.SYS2,DISP=SHR
//SOURCE   DD   DSN=DFRMM.MASTER.CDS,DISP=SHR
//SYSIN    DD   *
 REPRO INFILE(SOURCE) OUTFILE(OUTS2) FROMKEY(P) TOKEY(P)
//*************************************************************
//**             SOURCE TO TWO TARGETS                       **
//** COPY "V" RECORDS BY VOLSER RANGE 100000-199999  TO SYS1   **
//** COPY "V" RECORDS BY VOLSER RANGE A00000-A00999  TO SYS2   **
//** COPY RACK RECORDS FOR THIS RANGE OF VOLUMES AS WELL       **
//*************************************************************
//COPYV    EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//OUTS1    DD   DSN=DFRMM.NEW.SYS1,DISP=SHR
//OUTS2    DD   DSN=DFRMM.NEW.SYS2,DISP=SHR
//SOURCE   DD   DSN=DFRMM.MASTER.CDS
REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'E500F1F0') COUNT(100000)
 /* copy volumes from V 100000       to  V  199999        */
 REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'C500F3F4F8F040404040F1F0') -
        TOKEY(X'C500F3F4F8F040404040F1F9F9F9F9F9')
 /*  COPY racks from E 3480    100000       to  E  3480    199999 */
 REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'C600F3F4F8F040404040F1F0') -
        TOKEY(X'C600F3F4F8F040404040F1F9F9F9F9F9')
 /*  COPY racks from F 3480    100000       to  F  3480    199999 */
 REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
      FROMKEY(X'E400F3F4F8F040404040F1F0') -
        TOKEY(X'E400F3F4F8F040404040F1F9F9F9F9F9')
 /*  COPY racks from U 3480    100000       to  U  3480    199999 */
 REPRO INFILE(SOURCE) OUTFILE(OUTS2) -
      FROMKEY(X'V 00C1F0') COUNT(1000)
```

Archived

```
  /*  COPY from V   A00000     to  V   A00999        */
 REPRO INFILE(SOURCE) OUTFILE(OUTS2) -
       FROMKEY(X'C500C3C1D9E3E2404040C1F0') -
         TOKEY(X'C500C3C1D9E3E2404040C1F0F0F9F9F9')
 /*  COPY racks from E CARTS   A00000       to  E  CARTS   A00999 */
 REPRO INFILE(SOURCE) OUTFILE(OUTS2) -
       FROMKEY(X'C600C3C1D9E3E2404040C1F0') -
         TOKEY(X'C600C3C1D9E3E2404040C1F0F0F9F9F9')
 /*  COPY racks from F CARTS   A00000       to  F  CARTS   A00999 */
 REPRO INFILE(SOURCE) OUTFILE(OUTS2) -
       FROMKEY(X'E400C3C1D9E3E2404040C1F0') -
         TOKEY(X'E400C3C1D9E3E2404040C1F0F0F9F9F9')
 /*  COPY racks from U CARTS   A00000       to  U  CARTS   A00999 */
//*************************************************************
//**              SOURCE TO TWO TARGETS                      **
//** COPY BIN RECORDS FOR STORE VLT1  TO SYS1                 **
//** COPY BIN RECORDS FOR STORE VLT2  TO SYS2                 **
//*************************************************************
//COPYB    EXEC PGM=IDCAMS,REGION=5000K
//SYSPRINT DD   SYSOUT=*
//OUTS1    DD   DSN=DFRMM.NEW.SYS1,DISP=SHR
//OUTS2    DD   DSN=DFRMM.NEW.SYS2,DISP=SHR
//SOURCE   DD   DSN=DFRMM.MASTER.CDS,DISP=SHR
//SYSIN    DD   *
REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
       FROMKEY(X'D9E4E5D3E3F140404040C3C1D9E3E2404040F0F0F0F0F0F0') -
         TOKEY(X'D9E4E5D3E3F140404040C3C1D9E3E2404040F0F0F0F9F9F9')
 /*  COPY bins from RUVLT1    CARTS   000000 for 1000 bins */
 REPRO INFILE(SOURCE) OUTFILE(OUTS1) -
       FROMKEY(X'E2E4E5D3E3F140404040C3C1D9E3E2404040F0F0F0F0F0F0') -
         TOKEY(X'E2E4E5D3E3F140404040C3C1D9E3E2404040F0F0F0F9F9F9')
 /*  COPY bins from RUVLT1    CARTS   000000 for 1000 bins */
 REPRO INFILE(SOURCE) OUTFILE(OUTS2) -
       FROMKEY(X'D9E4E5D3E3F240404040F3F4F8F040404040F0F0F0F0F0F0') -
         TOKEY(X'D9E4E5D3E3F240404040F3F4F8F040404040F0F0F1F9F9F9')
 /*  COPY bins from RUVLT1    3480    000000 for 2000 bins */
 REPRO INFILE(SOURCE) OUTFILE(OUTS2) -
       FROMKEY(X'E2E4E5D3E3F240404040F3F4F8F040404040F0F0F0F0F0F0') -
         TOKEY(X'E2E4E5D3E3F240404040F3F4F8F040404040F0F0F1F9F9F9')
 /*  COPY bins from RUVLT1    3480    000000 for 2000 bins */
//*************************************************************
//**              EACH TARGET CDS                            **
//** RUN EDGUTIL WITH   PARM='MEND' TO UPDATE THE NEW TARGET  **
//**   CDS TO HAVE THE CORRECT COUNTS FOR RACK/BINS.          **
//**   ALSO SETS OWNER VOLUME COUNTS AND BUILDS OWNER VOLUME  **
//**   INFORMATION, AND DELETES UNUSED DATA SET RECORDS       **
//*************************************************************
//MEND1    EXEC PGM=EDGUTIL,PARM='MEND'
//MASTER   DD   DISP=SHR,DSN=DFRMM.NEW.SYS1
```

Archived

```
//SYSPRINT DD   SYSOUT=*
//MEND2    EXEC PGM=EDGUTIL,PARM='MEND'
//MASTER   DD   DISP=SHR,DSN=DFRMM.NEW.SYS2
//SYSPRINT DD   SYSOUT=*
//*************************************************************
//**              EACH TARGET CDS                            **
//** RUN EDGUTIL WITH   PARM='VERIFY(ALL)' ON THE NEW TARGET  **
//**   CDS TO VERIFY THE INTEGRITY OF THE NEW CDS.            **
//*************************************************************
//VERIFY1  EXEC PGM=EDGUTIL,PARM='VERIFY(ALL)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   SUBSYS=(BLSR,'DDNAME=MASTERB')
//MASTERB  DD   DISP=SHR,DSN=DFRMM.NEW.SYS1
//* Use of Batch LSR can speed the running of EDGUTIL
//* IF you have not got BLSR setup on your system you must replac
//* the MASTER DD statement with the following one, and delete
//* the MASTERB DD statement
//*MASTER  DD   DISP=OLD,DSN=DFRMM.NEW.SYS1,
//*        AMP=('BUFND=50,BUFNI=50')
//SYSIN    DD   DUMMY
//VERIFY2  EXEC PGM=EDGUTIL,PARM='VERIFY(ALL)'
//SYSPRINT DD   SYSOUT=*
//MASTER   DD   SUBSYS=(BLSR,'DDNAME=MASTERB')
//MASTERB  DD   DISP=SHR,DSN=DFRMM.NEW.SYS2
//* Use of Batch LSR can speed the running of EDGUTIL
//* IF you have not got BLSR setup on your system you must replace
//* the MASTER DD statement with the following one, and delete
//* the MASTERB DD statement
//*MASTER  DD   DISP=OLD,DSN=DFRMM.NEW.SYS2,
//*        AMP=('BUFND=50,BUFNI=50')
//SYSIN    DD   DUMMY
```

Example 13-14 shows you an example of how you can create one back up procedure that works on both systems.

Example 13-14 Sample JCL to back up the DFSMSrmm CDS and JOUNAL

```
//BACKCDS   PROC ID=
//BACKUP    EXEC PGM=EDGBKUP,PARM='BACKUP'
//SYSPRINT  DD   SYSOUT=*
//MASTER    DD   DSN=DFRMM.MASTER.SYS&ID.,DISP=SHR
//BACKUP    DD   DSN=DFRMM.MASTER.BACKUP&ID.(+1),DISP=(,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(6233,(1000,999),RLSE),
//          DCB=(MODEL.VB9216,DSORG=PS)
//          PEND
//BACKS1    EXEC BACKCDS,ID=1
//BACKS2    EXEC BACKCDS,ID=2
```

Archived

13.1.5 Converting to an existing CDS

When you are converting information to DFSMSrmm, and want the target to be an existing CDS, you are really merging databases. We recommend that you follow the normal conversion process (as described in the conversion Redbooks) to the point where the control data set is built and verified, and you have started DFSMSrmm and executed any commands necessary to complete the conversion. At that point you know the conversion is good, and that you have a valid and complete CDS. You can now use the process described in 13.1.3, “Merging the CDS” on page 572 to merge two DFSMSrmm CDSs.

If you have identified duplicate records, you may be able to change the extract and conversion control statements to avoid those duplicates rather than taking the actions recommended in 13.1.1, “Considerations” on page 562.

13.2 Merging DFSMSrmm environments

This section discusses the following aspects of moving a system using DFSMSrmm into a sysplex where all the existing systems share a single DFSMSrmm database:

- The necessity of merging DFSMSrmm environments, or possibility of having more than one DFSMSrmm environment in a single sysplex?
- A checklist of things to consider should you decide to merge the DFSMSrmms
- A methodology for performing the merge
- A list of tools and documentation that can assist with this exercise

While some of the concepts in this chapter probably apply to tape management systems other than DFSMSrmm, this chapter has been written specifically for DFSMSrmm by people familiar with that product. Correspondingly, the term we use in this chapter and in the index to describe systems sharing a single tape management system database is *RMMplex* (as opposed to TAPEplex or TMSplex, or similar terms).

13.2.1 One or more RMMplexes

It *is* possible to have more than one RMMplex in a single sysplex. By definition, the RMMplex is the set of systems that share a single DFSMSrmm Control Data Set (CDS).

If your target environment is a BronzePlex, the incoming system may share tape drives with systems in the target sysplex, but it will not share any DASD other

than the volumes containing the CDSs. If it is not sharing catalogs and the DFSMSrmm database (at a minimum, and probably also the RACF database and SMS environment) then the incoming system will not be part of the same RMMplex as the target sysplex.

If your target environment is a GoldPlex, the incoming system will be sharing the system volumes (and therefore, potentially, the volume containing the DFSMSrmm database) with the other systems in the target sysplex. However, because all the user catalogs and RACF and so on are *not* shared, you will once again probably not be merging your RMMplexes.

Finally, if your target environment is a PlatinumPlex, the incoming system *will* be sharing the user DASD (and RACF and SMS and so on) with the other systems in the target sysplex. In this case, you *would* normally merge the RMMplexes.

13.2.2 Considerations for merging RMMplexes

This section contains (in checklist format) a list of elements that must be considered should you decide to merge two or more RMMplexes. We say *RMMplexes* rather than *DFSMSrmm CDSs*, because there are more things to be considered than just merging the contents of the DFSMSrmm CDSs.

Due to the interrelationship between the following, it is strongly recommended that you merge the SMS environment, the HSM environment, the security environment (for example, RACF), the catalog environment, and the DASD environment, at the same time that you merge the DFSMSrmm environments.

Table 13-1 Considerations for merging RMMplexes

Considerations	Note	Type	Done
Ensure toleration service is installed if necessary	**1**	**P**	
Identify and eliminate duplicate records in the DFSMSrmm CDSs	**2**	**P**	
Review Parmlib options	**3**	**P**	
Review DFSMSrmm Procedure	**4**	**P**	
Review DFSMSrmm CDS and Journal data set sizes, and attributes	**5**	**P**	
TCDB Tape CDS	**6**	**P**	
Review any housekeeping jobs	**7**	**P**	
Security Considerations	**8**	**P**	

Archived

Review DFSMSrmm exits	**9**	**P**	

The Type specified in Table 13-1 relates to the sysplex target environment: **B** represents a BronzePlex, **G** represents a GoldPlex, and **P** represents a PlatinumPlex. Any notes concerning Table 13-1 are described below.

1. Ensure toleration service is installed if necessary.

   Before you can start a DFSMSrmm merge process, check the DFSMSrmm software levels. If the DFSMSrmm software levels are different, then you should check for compatibility PTFs, and confirm that they are installed and applied.

   To get a list of toleration and compatibility APARs for DFSMSrmm, refer to the DFSMS program directory, the DFSMSrmm PSP bucket, and Information APAR II08878.

2. Identify and eliminate duplicate records within the DFSMSrmm CDSs.

   Note: There are numerous references in this section to duplicate records and how to handle them. However, the first step has to be to *identify* whether you have duplicate records. The easiest way to do this is to run the jobs in Example 13-25 on page 603 and Example 13-26 on page 603. The description of the first step of the job, “Step S001ICET” on page 599, discusses how the merge job handles the various record types, and specifically, what it does with any duplicate records that it finds. You should look in the data sets containing the duplicate records to see which record types you need to address.

   Figure 13-1 contains a graphical summary of the process of merging multiple RMM CDSs. In this example, each LPAR has its own CDS, whereas in our sysplex merge, we will (hopefully) only have two: one from the incoming system and one from the target sysplex.

Archived

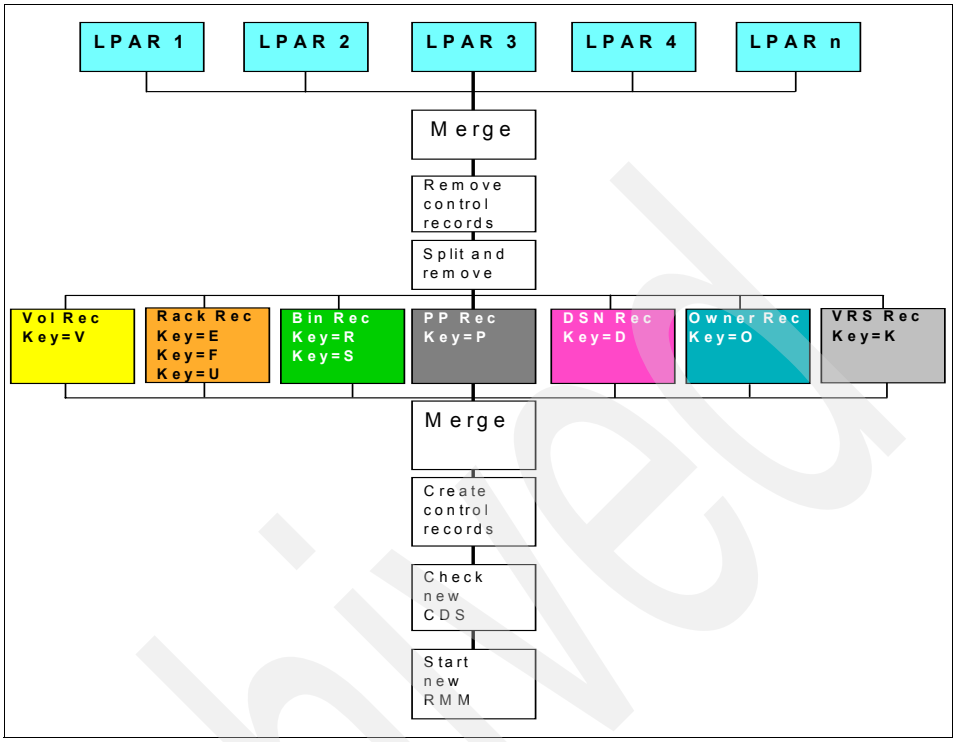


Figure 13-1 Merging process for multiple RMM CDSs

When merging CDSs, it is important to understand the contents of the CDS. For example, if you are merging two CDSs, you cannot merge them if the same resource is defined in each CDS. You must analyze the contents of the CDS to ensure that there are no duplicate resources (or at least, no conflicting duplicates).

If you find that there is a large number of duplicate tape volsers, then you should consider whether you really want to merge your DFSMSrmm CDSs. If you have a large number of duplicate volumes, and you would still like to merge the systems, then before you perform the merge, you should consider using one of the many TAPECOPY utility programs available in order to copy the duplicate volumes to volumes with unique volsers.

A sample merge job is shown in Example 13-26 on page 603. In addition to performing the merge, the job can also be used to identify duplicate records in advance of the actual merge. The job can be continually rerun until all duplicates (or at least those that require action) have been resolved. The merge job is described in more detail in “Merging the CDSs” on page 599. The merge job uses **REPRO** for merging the system’s DFSMSrmm CDSs into a flat file, sorts the file, discarding duplicate records in the process, and creates

a new merged DFSMSrmm CDS. Duplicate records are written to sysout files for you to investigate and take action. Note that all the duplicates that are deleted are from the same system. If you plan to use the merge process to remove duplicate records, all of the records that you want to discard *must* come from the same system; for example, you cannot use the merge to discard VRS records from RMMPLEX1 and volume records from RMMPLEX2.

CDS record types

In this section, we describe each resource type in the CDS, and explain what you can do to handle duplicate records.

The record types contained in the DFSMSrmm CDS are:

Key type	Record type
C	Control
CA	Action
CM	Move
D	Data set
E	Empty rack
F	Scratch rack
K	VRS
O	Owner
P	Product
R	Empty bin
S	In-use bin
U	In-use rack
V	Volume

Volume record

The volume record contains detailed information about a volume and its contents. It also identifies the data sets that are on the volume and volume chaining information. Some fields in the records identify the existence of other records in the CDS.

When moving volume records from one CDS to another, you must be sure to move the related records as well. These include the data set records, owner record, rack record, bin records, and product record. If you do not move them, the EDGUTIL utility may not be able to identify all the missing records, and therefore may not be able to create all the missing records.

When you are merging CDSs, you should not have any duplicate volume serials. In fact, from an integrity point of view, you should not have duplicate volume serials in the same sysplex in any case, regardless of whether or not it causes a problem in DFSMSrmm. If you do, you must remove the duplicate volumes

Archived

before starting the merge process. You should understand why there are duplicate volumes, and based on your understanding, select the best approach for dealing with them. While the merge job will fix some duplicate record problems by simply discarding one of the duplicates, you *must* ensure that there are no duplicate volume records before you start the merge. When you run the merge job, duplicate volume records are written to the file specified on the `DUPSMAIN DD` statement.

If you have duplicate volumes, there could also be duplicate data set records for those volumes. If you delete the volumes, the related duplicate data set records will be deleted automatically as part of the process of removing the duplicate volume.

Control record

Each DFSMSrmm CDS contains a control record. The control record contains the dates and times of when the inventory management functions last ran, the counts of rack numbers and bin numbers for built-in storage locations, and the settings for some DFSMSrmm options.

When merging databases, only one control record is required. You must use the EDGUTIL program with PARM=MEND to correct the record counts after the merge job has completed, and before you start DFSMSrmm using the new CDS.

The control record also contains the CDSID value. When merging the databases, retain the control record from the system that contained the CDSID you wish to continue to use (probably the one from DFSMSrmm on the target sysplex). All systems that use the DFSMSrmm CDSs must have a matching CDSID specified in the EDGRMMxx member of PARMLIB before being allowed to use the CDS.

As an alternative, you can bypass copying all control records, and after you have loaded the new DFSMSrmm CDS, you can create a new control record by using EDGUTIL with PARM=CREATE. After this has completed, use the same program with PARM=MEND to update the control record to correct the rack and bin counts.

You should use the RMM **`LISTCONTROL CNTL`** subcommand to check the options which are in use, such as EXTENDEDBIN and STACKEDVOLUME. If the options in use differ between the CDSs, we recommend that you enable options to bring the CDSs in line with each other, and follow the steps to implement those options *before* starting the merge.

Action and Move records

Action records contain information about the librarian actions that are outstanding. They include the release actions and any volume movements that are required. Duplicate records will be deleted by the merge job, because there is

a strong chance that they exist in each CDS, and the records will be rebuilt by DFSMSrmm during inventory management after the merge.

Data set record

A data set record exists for each volume on which the related data set has been written. Each record is uniquely identified by the volser and physical file number of the data set. Remember that it is perfectly acceptable to have multiple tapes all containing the same data set (unlike DASD data sets where you generally would try to avoid having the same data set name on more than one volume).

When merging CDSs, you want to be sure that there are no duplicate records across the two CDSs. Because the key of the data set record includes the volser, avoiding duplicate volsers ensures that there will be no duplicate data set records.

VRS record

The VRS records contain the retention and movement policies for your data. When merging the CDSs, you may find that you have the same policies defined in each CDS for some types of data. The policies are identified by fully qualified or generic data set name and job name, volser, and volser prefix.

When you merge CDSs, you must check for and resolve any duplicate policies. To help you do this, use the RMM TSO **SEARCHVRS** subcommand to get a list of all policies. If you have a policy for the same data set name and job name combination in more than one CDS, you should check that the retention and movement defined are the same. If they are the same, you can use the policy from any CDS. If they are not the same, you must resolve the conflict before merging the CDSs; this way, duplicates can be ignored, regardless of which system the records are taken from.

The most important reason for checking the VRS definitions is to check which VRS policy will apply to a given data set and jobname combination, and what impact this could have after the merge. If you use generic VRS definitions (like TOTHSM.**), it is possible that the VRS definition controlling a given data set and job could change even if you do not have duplicate policy names, potentially resulting in your keeping fewer generations of that data set/job than you had intended. This is similar to the discussion in the RACF chapter about data sets being protected by different RACF profiles after a merge. You should check all the VRS definitions for potential clashes between VRS definitions; if you find a clash, check, and, if necessary, modify the attributes of the VRS to meet your requirements.

There is a way to check whether the VRS records in the merged CDS will provide the results you expect. You can run EDGHSKP with PARM='VRSEL,VERIFY' using the test CDS that is created by the merge job. Specifying this PARM results

in a trial run on inventory management. The resulting ACTIVITY file will contain a list of all changes that will be made. By using the EDGJACTP sample job and looking at the summary of changes in matching VRS, you can see if in fact you would have this problem.

Owner record

There is one owner record for each identified user that owns a DFSMSrmm resource such as a volume or VRS. There can also be owner records for users that own no resources. Perhaps you have defined these so that the information can be used within your installation.

The owner record contains detailed information about the user and the count of volumes owned. For users who own volumes, DFSMSrmm uses multiple owner records to provide directed retrieval of volume and data set information during search commands. You only need to copy the “Base” Owner records. When there are multiple owner records, there is no need to copy the Owner records that contain just volume information. Once the records are loaded into the new CDS, your merge job will use the EDGUTIL utility with the PARM=MEND option to set the correct counts of owned volumes and build the owner volume records for owned volumes. A sample is shown in Step S004MEND in the “DFSMSrmm Merge job” in Example 13-26 on page 603.

Duplicate owner records are not a problem if the duplicate owners are the same user. In this case, duplicates can be safely ignored, regardless of which system the records are taken from.

If the duplicate owners are different users, you must decide how to handle that situation. You may have to assign different user IDs or define a new owner ID and transfer the owned resources to the new owner ID. Your objective is to have no duplicate owners, or to have any duplicates be for the same user in each case.

To add a new owner, you would issue the following command you can see in Example 13-15 on the RMMplex that the owner is being changed on.

Example 13-15 Create a new owner

```
RMM ADDOWNER new_owner_id DEPARTMENT(MERGE)
```

To delete the duplicate owner and transfer volume ownership to the new owner ID (after you have added the new owner ID), use the following command as shown in Example 13-16 on the same RMMplex as the **ADDOWNER** command:

Example 13-16 Delete owner and move all volume to a new owner

```
RMM DELETEOWNER old_owner_id NEWOWNER(new_owner-id)
```

Product record

A product record identifies a single product and software level and the volumes associated with that product. When you are merging CDSs, the existence of duplicate product records can be resolved by running EDGUTIL MEND after the merge.

As an alternative to relying on MEND to resolve any duplicates, you can resolve the duplicates before the merge by choosing one of the product names to change. First, list all the volumes associated with the product name, using a command similar to Example 13-17.

Example 13-17 List program product record

```
RMM LISTPRODUCT oldproduct
```

Then create a new product name using the **ADDPRODUCT** command similar to Example 13-18.

Example 13-18 Add a new program product

```
RMM ADDPRODUCT newproduct LEVEL(V01R01M00) NAME(.....)     -
    DESCRIPTION(.....) OWNER(....)
```

Then change all the volumes that were associated with the old product name to the new product name using a **CHANGEVOLUME** command as shown in Example 13-19.

Example 13-19 Change the volumes product assignment

```
RMM CHANGEVOLUME vvvvvv NUMBER(newproduct) FEATCD(A001) LEVEL(V01R03M00)
```

Finally, you should delete the old product name using the **DELETEPRODUCT** command, as shown in Example 13-20.

Example 13-20 Delete old product record

```
RMM DELETEPRODUCT oldproduct LEVEL(V01R01M00)
```

If you do not resolve the duplicates before the merge, there will be duplicate product records during the merge, and all the ones in the second CDS specified on the `EXTRACT DD` statement will be discarded during the merge; the discarded records are written to the file specified on the `DUPSMAIN DD` statement. Therefore, if you decide to leave some duplicate records in the CDS, so they can be deleted

during the merge, you must make sure that all those records are in the same CDS. After the merge, running EDGUTIL MEND will clean up the associations between the Product and Volume records.

Rack numbers record

There is a single rack record for each shelf location you have defined to DFSMSrmm. Rack records are optional when the rack number and the volume serial number have the same value. There may be one for each volume defined, but there can be more if you also defined empty rack numbers for use when volumes are added to the CDS. The rack number represents the volume's external volser.

The DFSMSrmm record for each rack number can begin with one of three keys: E for empty, F for full, and U for in-use. So, it is possible that you can have duplicate rack numbers in the two CDSs without the merge job spotting them as duplicates. For example, rack number 200000 in one DFSMSrmm CDS could be Empty (so the key would be E200000), and rack number 200000 in the DFSMSrmm CDS could be Full (so the key would be F200000) but the merge job ignores the key when identifying duplicate records so duplicate rack numbers *will* be identified.

When you merge CDSs, you must not have any duplicate rack numbers. Any duplicates must be deleted prior to the merge. Note that if there are duplicate records, the "U" records will be selected ahead of the "F" records, and finally the "E" records. Duplicates will be discarded, regardless of which CDS each record may have resided in prior to the merge. Any duplicates found at the time of the merge will be written to the file specified on the `DUPSEFU DD` statement.

Each rack number must have only one record type: either E, F, or U; you cannot have two record types for an individual rack number. If the duplicate already contains a volume, you can reassign the volume to another rack number by using the RMM **`CHANGEVOLUME`** subcommand. There are two different ways of reassigning a volume to a new rack number as shown in Example 13-21.

Example 13-21 Change the assigned rack number

```
RMM CHANGEVOLUME PP1234 POOL(A*)          or,
RMM CHANGEVOLUME PP1234 RACK(AP1234)
```

In the first command, DFSMSrmm picks an empty rack in pool A*. In the second example, you have selected the empty rack to be used. Remember that the rack number is the external label on the tape, so changing large numbers of rack numbers is not a trivial exercise. It helps if you can select a new rack number that is similar to the old one (for example, we changed rack number PP1234 above to rack number AP1234). Before cleaning up any duplicate rack numbers, refer to

the description of the “Volume record” on page 569 for an explanation of how to handle duplicate volumes.

If you currently have rack numbers for volumes when the rack number and the volume serial number have the same value, you can choose to stop using rack numbers before starting the merge. This would reduce the numbers of records in the CDS. To change a volume to no longer use a rack number, you can use the commands as shown in Example 13-22.

Example 13-22 Change volume to norack and delete the rack

```
RMM CHANGEVOLUME 100200 NORACK
RMM DELETERACK 100200
```

Bin numbers record

There is a single bin record for each storage location shelf location you have defined to DFSMSrmm.

Like the rack numbers we just discussed, each bin number record can potentially have one of two different keys; in this case, R (empty) and S (in-use). Because the same bin can be represented by two different keys, the merge jobs may not correctly identify all duplicates. For example, bin number 000029 can be represented by record R000029 in one CDS, and by record S000029 in the other. However, as for the rack numbers, the sample merge job is set up to ignore the key when searching for duplicate bin records, so duplicates will be identified (they are written to the file specified on the `DUPSRS DD` statement). Note that if there are duplicate records, the “S” records will be selected and the “R” records discarded, regardless of which CDS each record may have resided in prior to the merge.

When you merge CDSs, you must not have any duplicate bin numbers. Any duplicates must be addressed before you do the merge. If the duplicate already contains a volume, you can reassign the volume to another bin number by using the RMM **`CHANGEVOLUME`** subcommand. You can reassign a volume to a new bin number by issuing the following command as shown in Example 13-23.

Example 13-23 Reassign a bin number

```
RMM CHANGEVOLUME A00123 BIN(000029)
```

You can also remove the volume from the storage location, free up the bin number, and delete the unused bin number. The following commands show how to delete duplicate bin number X00022 in location VAULT1 of media name CARTS as shown in Example 13-24.

Example 13-24 Move a volume to its home location

```
RMM CHANGEVOLUME A00123 LOCATION(HOME) CMOVE
RMM DELETEBIN X00022 MEDIANAME(CARTS) LOCATION(VAULT1)
```

The next time you run inventory management, the volume is reassigned to the storage location, and another bin number is assigned.

Note: When you need to reassign bin numbers, you should consider how off-site movements are managed in your installation. If you depend on the volume being in a specific slot in a storage location, you should ensure that any volumes you change are physically moved in the storage location.

3. Review DFSMSrmm PARMLIB options

   To allow for a simplified backout process, it is recommended that you do not change the current DFSMSrmm PARMLIB member, but rather create a new merged version. On the actual merge day, you can simply rename the members to pick up the merged definitions.

   You should review the IEFSSNxx member, and confirm that the DFSMSrmm subsystem name and definitions are the same on all systems that will be sharing the same DFSMSrmm CDS.

   You should review the EDGRMMxx member for differences, and consolidate the members. Be aware of any changes you make that require supporting changes to RACF. You should also investigate whether all systems can share the same EDGRMMxx PARMLIB member as opposed to having a separate member for each DFSMSrmm instance.

   The EDGRMMxx options that are affected by the merging of CDSs are:

   – OPTIONS

     The options identify the installation options for DFSMSrmm. When you are merging CDSs, consolidate the **OPTIONS** command operands from all CDSs you are merging to ensure that you are using the same **OPTIONS** command operands on all DFSMSrmm subsystems.

     After you merge DFSMSrmm CDSs, the CDSID of at least one DFSMSrmm subsystem will probably change. You must update the CDSID parameter to match the new CDSID, or if you have created a new DFSMSrmm CDS control record and you have not specified a CDSID, DFSMSrmm creates the ID in the control record from the CDSID automatically.

     If the DFSMSrmm CDSs are kept in synch with the catalogs on the system, and you have enabled this using the inventory management CATSYNCH parameter, you should review the CATSYSID values specified

for each system. If all systems sharing the merged DFSMSrmm CDS will also have access to the same user catalogs for tape data sets (which should be the case), you can use CATSYSID(*). However, if the catalogs are not fully shared, you must produce a consolidated list of systems IDs for use with the CATSYSID and set the correct values in EDGRMMxx for each system. Before you use the merged CDS, we recommend that you mark the CDS as not in synch with the catalogs. Use the EDGUTIL program with UPDATE and the SYSIN statements with CATSYNCH(NO) to do this. Once you have started DFSMSrmm using the newly merged CDS, you can re-enable catalog synchronization using inventory management as detailed in the *DFSMSrmm Implementation and Customization Guide,* SC26-7405.

If you use the VRSMIN option to specify a minimum number of VRSs, remember to update the value if you have a different number as a result of the merge.

– LOCDEF

These options identify your locations to DFSMSrmm. When you are merging CDSs, consolidate the LOCDEF options from all CDSs you are merging to ensure that you have no duplicates, and that no locations are left out. If you have duplicates, ensure that they are identified as the same location type and with the same media names.

– VLPOOL

These options identify the ranges of rack numbers and pools of shelf space you have in your library. If your library is not changing, we recommend that you do not change your current VLPOOL options.

If you are merging CDSs and your existing VLPOOL definitions are different for each existing CDS, you must merge the definitions together to cover the merged sets of volumes and rack numbers. Ensure that the operands specified on each VLPOOL are correct when the same pool prefix is defined in more than one source environment.

– REJECT

These options identify the ranges of volumes to be rejected for use on the system. The reject specifies the rack number prefix, or if a volume has no library shelf location, DFSMSrmm checks the volume serial number. The range of volumes you specify on the `REJECT` statement should match the VLPOOL definitions that you have defined in RMM. For example, if you currently specify that the incoming system can only use volumes beginning with A*, and the target sysplex systems can only use volumes beginning with B*, you will need to update the `REJECT` statement at the time of the merge to indicate that all those systems can use tapes starting with either A* *or* B*.

Where you have a tape library shared by multiple systems, define the same VLPOOL definitions to all systems and use the REJECT definitions to prevent systems from using volumes that are for use on other systems.

The actions you take for the REJECT definitions should match how you handle the VLPOOL definitions.

The REJECT option can be used, for volumes defined to DFSMSrmm, to control the use of volumes on the system. And for undefined volumes, it can be used to control the partitioning of system managed libraries. Consider both of these aspects as you merge the REJECT definitions.

4. Review the DFSMSrmm started task JCL

   To allow for a simplified backout process, it is recommended that you do not change the current DFSMSrmm procedure, but rather create a new merged version. On the actual merge day, you can simply rename the two procedures to perform the merge.

   You should review the DFSMSrmm procedure and investigate whether all systems can share the same DFSMSrmm procedure. This should be possible with little or no work as the only system-specific information that is contained in the DFSMSrmm procedure is the names of the PDO (EDGPDOx) data sets. PDO data sets are optional, but if you do define them, they should be unique on each system; we suggest that you include the system name (&SYSNAME) in the data set name.

   The DFSMSrmm CDS and Journal names can be specified in either the DFSMSrmm JCL or in the PARMLIB member EDGRMMxx. For flexibility, we recommend that they are specified in EDGRMMxx; this allows you to easily change the CDS name by switching to a different PARMLIB member.

   You should also check that the REGION size specified for the DFSMSrmm started procedure is large enough; review the latest recommendations for this in the *DFSMSrmm Implementation and Customization Guide,* SC26-7405.

5. Review the DFSMSrmm CDS and journal data set sizes and attributes

   To allow for a simplified backout process, we recommend that you create new versions of the DFSMSrmm CDS and Journal data sets. On the actual merge day, you can simply rename the old ones, and rename the new ones to be ready to start DFSMSrmm after the merge**.**

   Before you create the new data sets, calculate the new combined CDS size, and update the merge job with this information so that the CDS allocated in that job is valid.

   At the same time, calculate the new journal data set size, and create a job to allocate the data set.

Archived

Review the sample allocation jobs for the DFSMSrmm CDS, member EDGJMFAL in SYS1.SAMPLIB and for the DFSMSrmm journal, member EDGJNLAL in SYS1.SAMPLIB to ensure that you are defining the new CDS with the latest information regarding record size, CISize, and Shareoptions. For more information about these data sets, refer to *DFSMSrmm Implementation and Customization Guide,* SC26-7405.

6. TCDB Tape Configuration data set

   If the systems to be merged contain Volume Entries cataloged inside either a General Tape Volume Catalog (SYS1.VOLCAT.VGENERAL, for example) or a specific Tape Volume Catalog (SYS1.VOLCAT.Vx, for example), you will need to list all the volume entries and create the volume entries in the target TCDB with the same attributes as in the current system. You can use a REPRO MERGECAT to copy entries from the old to the new volume catalogs if required.

   You will also need to confirm that the SMS ACS routines are allocating tape data sets to the appropriate storage groups associated with the tape libraries.

   Review the DEVSUPxx member in SYS1.PARMLIB if the TCDB is being merged and make sure that all systems are entering the tapes with the same category number.

   If the category number changes for any systems, there is no need to make any changes for private volumes since when the volume returns to scratch, its category will be changed to one of the new scratch categories. However, you will need to make changes for volumes already in Scratch status. There is more information about this in the section entitled “Processing Default Categories when using DEVSUPxx in an ATLDS” in *DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Tape Libraries,* SC35-0427.

   You will need to obtain a list of volumes whose storage group name is *SCRTCH* using the ISMF Option 2.3 Mountable Tape. Use the **ISMF ALTER** command (not the line operator) to change the volume use attribute for all volumes in the list from scratch to scratch; this causes the library manager category for each volume to be changed to the new value established through DEVSUPxx.

7. Review any housekeeping jobs

   Review any housekeeping jobs, confirming that data set names are valid for the new merged environment. It should be possible to eliminate any DFSMSrmm jobs that were running on the incoming system previously, except for any report jobs that were not running on the target sysplex. If you have switched on the catalog synchronization feature, you must check that the CATSYSID definitions are correct and that the inventory management

vital record processing is running on the correct system. Also, the inventory management expiration processing must be checked and running on multiple systems if not all user catalogs are shared (hopefully, in a PlatinumPlex *all* catalogs are shared between all systems).

8. Security considerations

   Chapter 8 in the *DFSMSrmm Implementation and Customization Guide,* SC26-7405, discusses security for DFSMSrmm. If the systems to be merged are also merging the RACF environment at the same time, and you have set up the PARMLIB definitions the same on all systems, there should be no additional security concerns. If either TAPEVOL or TAPEDSN are active, make sure all EDGRMMxx definitions for TPRACF are the same. If you have not switched on TAPEVOL or TAPEDSN on all systems before you have merged the DFSMSrmm CDS, but now you will use them on all systems sharing the DFSMSrmm CDS, you must define all needed resources before the merged DFSMSrmm is started.

9. Review DFSMSrmm exits

   You will need to determine what DFSMSrmm exits are in use (for example, EDGUX100, EDGUX200, CBRUXCUA, CBRUXENT, CBRUXEJC and CBRUXVNL). You should review the functions of these exits on the incoming system and the target sysplex systems. If they are not the same, you should see if you can create a single version of each exit so that all systems will be using identical exits, thus ensuring consistent processing.

13.2.3 Methodology for merging RMMplexes

In this section, we discuss the approaches that can be taken to merge RMMplexes, and give some advice about the approach that provides the smallest risk or is the easiest to implement.

An alternate approach to the one discussed within this book is to create **DFSMSrmm ADD....**commands for all the volumes, data sets, racks, bins, and vital record specifications being merged. This approach can be used to avoid any outage to DFSMSrmm. Care must be taken to ensure that the appropriate security options are already in place, for example, TAPEVOL/TAPEDSN(TVTOC). Since there are no tools available to do this, you need to create these yourself.

The approach discussed in this document uses Repro for the merging system's DFSMSrmm CDSs, sorts the resulting flat file, and creates a new merged DFSMSrmm CDS. Duplicate and unnecessary records are not included in the new CDS. Duplicates are written to sysout files for your review, and to confirm that you do not require them. DFSMSrmm is then restarted on all systems using the new merged DFSMSrmm CDS.

Merging the CDSs

Follow these steps to merge the CDSs:

1. Prior to the day of the merge, you should have addressed all the issues and tasks identified in 13.2.2, “Considerations for merging RMMplexes” on page 584.
2. Stop DFSMSrmm on all systems in both RMMplexes. When you run the merge job, DFSMSrmm on all systems must be down. When you stop DFSMSrmm, do *not* use the OPT=RESET function, as it allows tapes to be processed without DFSMSrmm's knowledge. You will not be able to use any tapes from this time until the merge is complete.
3. Back up both sets of CDSs and Journals using the sample JCL in Example 13-25 on page 603. This JCL uses Repro to back up the files. While you should not change any of these files during the merge, we are taking the backups to protect you from any accidental damage. Also, by using Repro for the backup, we can use the backup files as input to the merge process.
4. If required, perform any merging of Volume Entries in the Target TCDB Tape Catalog Database.
5. Run the merge job shown in Example 13-26 on page 603, using the data sets created in Step 3.

   The sample merge job uses ICETOOL. If you do not have this program available, there is an alternate method using IDCAMS discussed in this chapter. If SYNCTOOL is available instead of ICETOOL, this job may run with little or no changes to the JCL and statements.

   The sample merge job can be run while RMM is running and while you are performing testing. However, when you are doing the actual merge, all RMM subsystems should be stopped before you run the merge job.

   The merge job consists of a number of steps. The steps, and the data sets used by each step are as follows:

   a. **Step S001ICET**

      The first step in the job simply deletes the output files that may have been left from the last run of this job.

   b. **Step S001ICET**

      The next step in the job uses the ICETOOL utility of DFSORT. The input data sets are sequential unloads of the two DFSMSrmm CDSs (created with Repro) and are specified on the `DD EXTRACT` statement. If duplicates are discovered in the EXTRACT files, the first occurrence will be retained and the second and subsequent ones will be discarded (with the exceptions previously noted for the rack and bin records). Therefore, the order of the DD statements on the EXTRACT is critical. The first DD

should refer to the flat file from the RMMplex whose CDSID you wish to retain. Whenever duplicate records are detected, the records from that RMMplex will be retained and those from the other RMMplex will be discarded. For this reason, you must be sure that any records that you want the merge job to discard are all in the same RMMplex.

The `MAINCNTL DD` statement contains the ICETOOL statements that will be used to sort and select most record types (C, K, P, and V). The associated `SELECT` statement in the `TOOLIN DD` specifies that duplicates should not be written the output file for these record types (MAIN1S), but should instead be written to DUPSMAIN for you to investigate and resolve.

The `EFUXCNTL DD` statement contains the `ICETOOL` statements that will be used to sort and select record types E, F, and U. They also ensure that the output file (EFU1S) will have only one record type for each rack. If the same rack number is discovered in more than one record type, records will be selected with the “U” types first, followed by the “F” types, and finally the “E” types. Duplicate records will be written to DUPSEFU for you to investigate and resolve.

The `RSXXCNTL DD` statement contains the `ICETOOL` statements that will be used to sort and select record types “R” and “S” into the data set specified on the RS1S DD statement. They also make sure you have only one record type for each bin. If the same bin number is discovered in more than one record type, the “S” record will be selected, and the “R” record will be written to DUPSRS for you to investigate and resolve.

The `OOOOCNTL DD` statement contains the `ICETOOL` statements that will be used to sort and select the type O records into the data set specified on the OO1S DD statement, and to make sure you have only one record for each owner. It will also discard type O records that just contain volume information (refer to “Owner record” on page 590).

Finally, the `DDDDCNTL DD` statement contains the `ICETOOL` statements that will be used to sort and select all the data set records. If you have addressed any duplicate volumes, there should be no issues with copying all data set records, as the key includes the volser and the physical file number, therefore, we have not provided a `DUPSD DD` statement as a destination for duplicate records.

c. Step S002ICET

This step also uses the ICETOOL Utility of DFSORT. The input data sets are the temporary data set created by S001ICET that contain the (non-duplicate) CDS records. An output data set (DD MERGED) will be created in the same format as the input data sets for S001ICET.

d. Step S003AMS

This step uses IDCAMS to define a new DFSMSrmm CDS, and REPROs the sequential file created in S002ICET (DD MERGED) into the new CDS. This is the CDS that will be used in the new merged RMMplex.

e. Step S004MEND

This step uses the EDGUTIL utility to repair all the record counts, set correct owners, and create missing records, as described previously. Running EDGUTIL with the MEND parameter only updates the CDS referred to on the MASTER DD card, so you can safely run this utility as many times as you like prior to the actual merge, *as long as you ensure that the MASTER DD points at your test CDS*.

Please note that if any of the volumes defined in the newly merged DFSMSrmm CDS are contained within an ATL or VTS, then the TCDB will need to already contain correct entries for these volumes, that is, SYS1.VOLCAT.VGENERAL will need to already have these volumes defined before the MEND can complete successfully. If all the volumes are inside an ATL or VTS, you can use the EDGUTIL MEND(SMSTAPE) function to add/correct the volumes in the TCDB.

Note: If you specify `SMSTAPE`, the TCDB will be updated, so you should not use this option until you do the actual merge.

f. Step S005MEND

Note that if there are many records to be fixed, there can be quite a lot of output from EDGUTIL. In order to get a clear picture of which records the MEND was unable to fix (and there should be *very* few of these), you should run the EDGUTIL program again, this time with the VERIFY parameter. This will identify any actions that MEND was not able to take, and therefore the ones you need to consider taking action for. The message you are most likely to see will be `"VOLUME xxxxxx NOT FOUND IN VOLUME CATALOG"`, and it will be fixed when you run MEND with the SMSTAPE option.

6. Run EDGUTIL with UPDATE to turn off CATSYNCH. It is necessary to turn off catalog synchronization before the merge is complete since the catalogs will need to be re-synchronized. Use the sample job in Example 13-27 on page 606.
7. Perform any renames of CDSs, Journals, PARMLIB members, started tasks and JCL.
8. Start the DFSMSrmm procedure using the new PARMLIB member.
9. Check that the merged information is valid and correct by using **`DFSMSrmm`** commands, and the ISPF dialog to retrieve and display information.

10. Run EDGHSKP with CATSYNCH if catalog synchronization is to be maintained by DFSMSrmm. We recommend doing this for the performance benefits it can have for inventory management when use is made of policies on catalog retention. This should be done before the first housekeeping job (EDGHSKP) is run after the merge.

11. If re-synchronization between the new merged DFSMSrmm CDS and TCDB is required, the DFSMSrmm EDGUTIL program should be run with a parameter of 'VERIFY(VOLCAT)'. In addition, if the systems in the RMMplex are all running OS/390 2.10 or later, you should also specify 'VERIFY(SMSTAPE)'.

12. Run DFSMSrmm inventory management on the new CDS. During this run, DFSMSrmm VRSEL and DSTORE processing identifies volumes to be moved to the correct location. If any moves are identified, use EDGRPTD to produce the picking lists and actions of any moves that you know are still required.

    The inventory management run should execute all inventory management functions including backup.

Backout

If backout is required at any stage, because you created new versions of the CDS, Journal, PARMLIB member and started task JCL, you can simply rename the old data sets back to the original names. There should be no problems as a result of the additional Volume Entries you may have created in the TCDB, but they will need to be cleaned up before re-attempting the merge process.

13.2.4 Tools and documentation

In this section, we provide information about tools and documentation that may help you complete this project.

Example 13-25 contains sample JCL that can be used to back up the DFSMSrmm CDSs and Journal data sets after the DFSMSrmm subsystems have been stopped. Note that this job will *not* reset the contents of the Journal data sets. Even though we will not be making any changes to any of these data sets during the merge process, it is prudent to take a backup of them before any changes are made, simply to protect yourself from any mistakes that might arise during the merge.

Archived

You can also use this JCL to create the input to the merge job for dummy runs in order to identify any duplicate records that may exist. Note, however, that if you run this JCL while the DFSMSrmm subsystems are running, you must comment out the //MASTER and //JOURNAL DD cards. If DFSMSrmm is *not* running when you run the EDGBKUP job, you will need to uncomment the MASTER and JOURNAL DD statements, and fill in the correct names for those data sets.

Example 13-25 JCL to back up the DFSMSrmm CDSs and journals

```
//SOO1BKUP EXEC PGM=EDGBKUP,PARM='BACKUP(NREORG)'
//SYSPRINT DD   SYSOUT=*
//*MASTER    DD  DISP=SHR,DSN=RMMPLEX1.RMM.CDS
//*JOURNAL   DD  DISP=SHR,DSN=RMMPLEX1.RMM.JOURNAL
//BACKUP    DD   DSN=hlq.RMMPLEX1.CDS.BACKUP,DISP=(,CATLG,DELETE),
//          DCB=(DSORG=PS,RECFM=VB,LRECL=9216),
//          UNIT=SYSALLDA,SPACE=(CYL,(xxx,xx),RLSE)
//JRNLBKUP  DD   DSN=hlq.RMMPLEX1.JRNL.BACKUP,DISP=(,CATLG,DELETE),
//          DCB=(DSORG=PS,RECFM=VB,LRECL=32756,BLKSIZE=32760),
//          UNIT=SYSALLDA,SPACE=(CYL,(xxx,xx),RLSE)
//SOO2BKUP  EXEC PGM=EDGBKUP,PARM='BACKUP(NREORG)'
//SYSPRINT  DD   SYSOUT=*
//*MASTER   DD   DISP=SHR,DSN=RMMPLEX2.RMM.CDS
//*JOURNAL  DD   DISP=SHR,DSN=RMMPLEX2.RMM.JOURNAL
//BACKUP    DD   DSN=hlq.RMMPLEX2.CDS.BACKUP,DISP=(,CATLG,DELETE),
//          DCB=(DSORG=PS,RECFM=VB,LRECL=9216),
//          UNIT=SYSALLDA,SPACE=(CYL,(xxx,xx),RLSE)
//JRNLBKUP  DD   DSN=hlq.RMMPLEX2.JRNL.BACKUP,DISP=(,CATLG,DELETE),
//          DCB=(DSORG=PS,RECFM=VB,LRECL=32756,BLKSIZE=32760),
//          UNIT=SYSALLDA,SPACE=(CYL,(xxx,xx),RLSE)
```

Example 13-26 contains the merge job that we have referred to and discussed earlier in this chapter. This job can also be used to identify duplicate records during the preparation for the merge.

Example 13-26 DFSMSrmm merge job

```
//CLEANUP EXEC PGM=IEFBR14
//DD1     DD   DSN=hlq.MERGED.RMM.MASTER.SEQ,DISP=(MOD,DELETE),
//        SPACE=(CYL,0),UNIT=SYSDA
//DD2     DD   DSN=hlq.MERGED.RMM.MASTER,DISP=(MOD,DELETE),
//        SPACE=(CYL,0),UNIT=SYSDA
/*
//SOO1ICET EXEC  PGM=ICETOOL
//TOOLMSG  DD   SYSOUT=*
//DFSMSG   DD   SYSOUT=*
//DFSPARM  DD   *
VLSHRT
/*
```

```
//DUPSMAIN DD   SYSOUT=*
//DUPSEFU  DD   SYSOUT=*
//DUPSRS   DD   SYSOUT=*
//DUPSO    DD   SYSOUT=*
//EXTRACT  DD   DISP=SHR,DSN=hlq.RMMPLEX1.CDS.BACKUP
//         DD   DISP=SHR,DSN=hlq.RMMPLEX2.CDS.BACKUP
//MAIN     DD   DSN=&&MAIN,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//EFU      DD   DSN=&&EFU,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//RS       DD   DSN=&&RS,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//OOOO     DD   DSN=&&OOOO,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//DDDD     DD   DSN=&&DDDD,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//MAIN1S   DD   DSN=&&MAIN1S,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//EFU1S    DD   DSN=&&EFU1S,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//RS1S     DD   DSN=&&RS1S,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//OOO1S    DD   DSN=&&OOO1S,REFDD=*.EXTRACT,SPACE=(CYL,(1,10),RLSE),
//         UNIT=3390,DISP=(,PASS)
//TOOLIN   DD   *
  SORT FROM(EXTRACT) TO(MAIN) USING(MAIN)
  SELECT FROM(MAIN) TO(MAIN1S) ON(5,44,CH) FIRST -
   DISCARD(DUPSMAIN)
  SORT FROM(EXTRACT) TO(EFU) USING(EFUX)
  SELECT FROM(EFU) TO(EFU1S) ON(15,6,CH) FIRST -
   DISCARD(DUPSEFU)
  SORT FROM(EXTRACT) TO(RS) USING(RSXX)
  SELECT FROM(RS) TO(RS1S) ON(15,6,CH) FIRST -
   DISCARD(DUPSRS)
  SORT FROM(EXTRACT) TO(OOOO) USING(OOOO)
  SELECT FROM(OOOO) TO(OOO1S) ON(5,44,CH) FIRST -
   DISCARD(DUPSO)
  SORT FROM(EXTRACT) TO(DDDD) USING(DDDD)
//MAINCNTL DD   *
  OPTION EQUALS
  SORT FIELDS=(5,44,CH,A)
  INCLUDE COND=(5,2,CH,NE,C'CA',AND,5,2,CH,NE,C'CM',AND,
                5,1,CH,NE,C'O',AND,
                5,1,CH,NE,C'D',AND,
                5,1,CH,NE,C'R',AND,5,1,CH,NE,C'S',AND,
                5,1,CH,NE,C'E',AND,5,1,CH,NE,C'F',AND,5,1,CH,NE,C'U')
/*
//EFUXCNTL DD   *
  OPTION EQUALS
```

```
  SORT FIELDS=(5,1,CH,D,15,6,CH,A)
  INCLUDE COND=(5,1,CH,EQ,C'E',OR,5,1,CH,EQ,C'F',OR,5,1,CH,EQ,C'U')
/*
//RSXXCNTL DD *
  OPTION EQUALS
  SORT FIELDS=(5,1,CH,D,15,6,CH,A)
  INCLUDE COND=(5,1,CH,EQ,C'R',OR,5,1,CH,EQ,C'S')
/*
//OOOOCNTL DD   *
  OPTION EQUALS
  SORT FIELDS=(5,1,CH,A)
  INCLUDE COND=(5,1,CH,EQ,C'O',AND,14,1,CH,EQ,X'00')
/*
//DDDDCNTL DD   *
  OPTION EQUALS
  SORT FIELDS=(5,1,CH,A)
  INCLUDE COND=(5,1,CH,EQ,C'D')
/*
//S002ICET EXEC  PGM=ICETOOL
//TOOLMSG  DD   SYSOUT=*
//DFSMSG   DD   SYSOUT=*
//DFSPARM  DD   *
VLSHRT
/*
//EXTRACT  DD   DISP=SHR,DSN=&&MAIN1S
//         DD   DISP=SHR,DSN=&&EFU1S
//         DD   DISP=SHR,DSN=&&RS1S
//         DD   DISP=SHR,DSN=&&OOO1S
//         DD   DISP=SHR,DSN=&&DDDD
//MERGED   DD   DSN=hlq.Merged.RMM.Master.Seq,
//         REFDD=*.EXTRACT,SPACE=(CYL,(xxx,yy),RLSE),
//         UNIT=SYSALLDA,DISP=(,CATLG)
//TOOLIN   DD  *
  SORT FROM(EXTRACT) TO(MERGED) USING(MOST)
//MOSTCNTL DD   *
  OPTION EQUALS
  SORT FIELDS=(5,56,CH,A)
/*
//S003AMS  EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD   SYSOUT=*,OUTLIM=1000000
//RMM      DD   DISP=SHR,DSN=hlq.Merged.RMM.Master.Seq
//SYSIN    DD   *
  DEFINE CLUSTER(NAME(hlq.Merged.RMM.Master) -
                 FREESPACE(20 20) -
                 KEYS(56 0) -
                 REUSE -
                 RECSZ(512 9216) -
                 SHR(3 3) -
                 CYLINDERS(xxx yy) -
```

Archived

```
                    VOLUMES(vvvvvv)) -
                DATA(NAME(hlq.Merged.RMM.Master.DATA) -
                    CISZ(10240)) -
               INDEX(NAME(hlq.Merged.RMM.Master.INDEX) -
                    CISZ(2048) -
                    NOIMBED -
                    NOREPLICATE)
     REPRO IFILE(RMM) ODS(hlq.Merged.RMM.Master)
  /*
  //*-------------------------------------------------------------------
  //SOO4MEND  EXEC PGM=EDGUTIL,PARM='MEND'
  //SYSPRINT  DD   SYSOUT=*
  //MASTER    DD   DISP=SHR,DSN=hlq.Merged.RMM.Master
  //SYSIN     DD   DUMMY
  //*-------------------------------------------------------------------
  //SOO5MEND  EXEC PGM=EDGUTIL,PARM='VERIFY'
  //SYSPRINT  DD   SYSOUT=*
  //MASTER    DD   DISP=SHR,DSN=hlq.Merged.RMM.Master
  //SYSIN     DD   DUMMY
```

Example 13-27 contains sample JCL that can be used to turn the CATSYNCH feature off after you have merged the CDSs, and before you start the DFSMSrmm subsystems using the new CDS.

Example 13-27 CATSYNCH job

```
  //EDGUTIL   EXEC PGM=EDGUTIL,PARM='UPDATE'
  //SYSPRINT  DD   SYSOUT=*
  //MASTER    DD   DISP=SHR,DSN=hlq.Merged.RMM.Master
  //SYSIN     DD   *
    CONTROL CATSYNCH(NO)
  /*
```

Finally, the following documents contain information that will be helpful as you proceed through this project:

- *DFSMSrmm Implementation and Customization Guide,* SC26-7405
- *DFSMSrmm Reporting,* SC26-7406
- *Converting to RMM - A Practical Guide,* SG24-4998
- *DFSMSrmm Primer,* SG24-5983

Archived

A

Conversion checklist

Use the checklist in Table A-1 to track the activities to be performed in order to achieve a successful conversion.

Table A-1 Conversion checklist

Task	Person	Plan start date	Actual start date	End date
Stage 1. Learning about the conversion				
Conversion process				
Learning about DFSMSrmm				
Stage 2. Planning for conversion				
Software maintenance levels				
Timing of events in your installation				
Change management				
Limiting business risk				
Preparing a conversion plan				

© Copyright IBM Corp. 2004. All rights reserved.

Task	Person	Plan start date	Actual start date	End date
Stage 3. Starting DFSMSrmm				
► **Stage 3a. Implementing DFSMSrmm**				
Update installation-wide exits				
Update and validate PARMLIB members				
Create the DFSMSrmm control data set (CDS)				
Create the DFSMSrmm journal				
Protecting DFSMSrmm resources				
Restart MVS with DFSMSrmm implemented				
Setting up DFSMSrmm utilities				
Running the installation verification program				
► **Stage 3b. Verifying the DFSMSrmm implementation**				
Display PARMLIB options and control information				
Add owner information				
Add racks to DFSMSrmm				
Add volumes to DFSMSrmm				
Add bins to DFSMSrmm				
Add a VRS to DFSMSrmm				
Create tape data sets				
Allocate inventory management data sets				
Run inventory management				
Confirm movements				
Create reports				

Archived

Task	Person	Plan start date	Actual start date	End date
Run EDGINERS to initialize volumes automatically				
Restore the CDS				
Verify the CDS				
Testing implemented exits				
Testing application use of tape				
► **Stage 3c. Education**				
Stage 4. Analyze current environment				
► **Stage 4a. Compare products**				
Terminology				
Functions				
Retention methods				
Data fields				
Cross-reference				
► **Stage 4b. Reviewing the current environment**				
Collecting information				
Analyzing the record layouts				
TLMS PARMLIB options				
Inventory management and reports				
Exits				
Data set overwrite				
BLP and NL processing				
DFSMShsm interface				
Tape initialization				
Tape pool management				
Security				

Archived

Task	Person	Plan start date	Actual start date	End date
Retention management				
Retention criteria				
Storage location management				
Movement policies				
Interfaces				
Stage 5. Extracting data for conversion				
Sample extraction programs				
Preparing to run the extract programt				
The EDGCDYNM program				
Stage 6. Completing conversion				
► **Stage 6a. Building the DFSMSrmm CDS**				
Preparing to run EDGCNVT				
EDGCNVT and EDGCEBIN processing				
EDGCEBIN processing				
EDGCNXIT user exit				
DFSMSrmm CDS creation				

Archived

Task	Person	Plan start date	Actual start date	End date
► **Stage 6b. Sample JCL for CDS create and verify**				
► **Stage 6c. Running DFSMSrmm commands**				
Adding empty bins				
Adding VRSs				
Stage 7. Parallel running and validation				
Preparing to run in parallel				
Test modifications to DFSMS ACS routines				
Parallel processing				
Starting tape and inventory management				
Validating DFSMSrmm operation				
Keeping databases synchronized				
Stage 8. Cutover to production				
EDGRMMxx PARMLIB options				
Control access to DFSMSrmm functions				
Activate exits				
Remove TLMS				
Modifying CA-Disk				
Inventory management				
Interface to tape libraries				
Starting DFSMSrmm				
Stage 9. Measuring success				
Final testing				
Commitment to DFSMSrmm				

Archived

Task	Person	Plan start date	Actual start date	End date
Stage 10. Full exploitation				
Uninstalling TLMS				
Cleaning up your DFSMSrmm implementation				
Implementing additional functions				

Archived

B

Conversion record layouts

The IBM-written EDGCNVT program requires a sequential file as input, and an optional file that includes customer-written policy and rules statements. This appendix provides details of the required record layouts for the sequential input file. There are five different records that describe input to EDGCNVT:

EDGCDREC	Maps the data set details record
EDGCEREC	Maps the empty shelf details record
EDGCKREC	Maps the policy details record
EDGCLREC	Maps the volume and first data set details record
EDGCOREC	Maps the owner details record

Archived

© Copyright IBM Corp. 2004. All rights reserved.

Record layout description

The description of the different five input records for the EDGCNVT program is shown in Table B-1.

Table B-1 Record layout description

Record Type	Description
L	Library record. An L-Record contains details of a single volume and the first data set on that volume.
D	Data set name record. A D-Record contains details of a single data set. One D-Record is required for each additional data set on a volume. If you supply a D-Record for a file, all previous files on the same volume must be specified in D-Records that precede the record. The L-Records and D-Records have the following sequence requirements: Library (L) records for a multi-volume set of volumes must appear in ascending volume sequence order, interrupted only by: Data set name (D) records, in ascending file sequence order, after each L-Record that is created for the volumes from which the data set originates. The count of D-Records that follow after a non-scratch L-Record must be one less than the count of data sets in the LVDSNCNT field of that L-Record, because each L-Record contains first file data set information.
E	Empty rack and bin record. The E-Record contains details of a single, empty shelf location. It can describe either a rack number or a bin number.
O	Owner. The O-Record contains details of a single owner ID. The owner may own resources that are being converted, but it need not own any resources.
K	VRS. The K-Record contains details of a single policy definition.

Archived

EDGCDREC record layout

```
              RMM CONVERSION DATA SET RECORD MAPPING
z/OS DFSMSrmm V1R3
PROPRIETARY V3 STATEMENT
LICENSED MATERIALS - PROPERTY OF IBM
"RESTRICTED MATERIALS OF IBM"
5694-A01
(C) COPYRIGHT IBM CORP. 1993,2001
END PROPRIETARY V3 STATEMENT
   FUNCTION : EDGCDREC MAPS THE CONVERSION DATA SET RECORD
   LOCATION : N/A
   METHOD OF ACCESS:
              ASSEMBLER - USING ON RMMDREC
              LA Rx,DRECAREA
                USING RMMDREC,Rx
              MVI DSNRID,C'D' Set D record type
              PLX - %INCLUDE EDGCDREC
 LISTING   :  ASSEMBLER - SPECIFY LIST=YES ON THE MACRO
              PLX - SPECIFY &IHALIST='YES' BEFORE INCLUDE

CHANGE ACTIVITY:
$00=OW0xxxx,110,940920,MWW: Create Macro
$01=OW11151,110,940921,MWW: Implement new macro
$02=OW12935,120,951004,GD: Add supporting PLX/370
$K1=K151038,150,980616,BDG: Correct spelling and flags
$03=OW32993,130,980607,RS : sync PL/X and Assembler part
$04=OW37600,140,990112,RS : Add SMS management class value
$05=OW40710,140,990819,RS : Add CREATING PROGRAM NAME
$LL=RMM213 ,213,010216,BDG: >9999 tape files
$06=OW56705,210,021004,MWW: BLKSIZE incorrect after conversion
$07=OA02095,213,030224,AK : Data set expiratoin date support
```

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
START OF RMMDREC					
0	(0)	STRUCTURE	208	CDREC	ZERO
0	(0)	CHARACTER		DREC	BOUNDARY ALIGN
0	(0)	SIGNED	4	DRECRDW	RESERVED FOR RDW

Arrived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
4	(4)	CHARACTER 11.. .1..	1	DSNRID DSNRTYPE	(REQUIRED) DATA SET NAME RECORD ID 'D' - DATASET NAME RECORD TYPE
5	(5)	CHARACTER	4	DDCRDATE	(REQUIRED) DSN CREATE DATE - YYYYDDD
9	(9)	CHARACTER	4	DDCRTIME	DSN CREATE TIME - HHMMSST
13	(D)	CHARACTER	44	DSNNAME	(REQUIRED) DATA SET NAME
57	(39)	BITSTRING 1111 .... 1... .... .1.. 11	1 * *	DSNTRTCH * DSNTIDRC DSBTNCOM	(REQUIRED) RECORDING TECHNIQUE TRTCH RESERVED DATA is compacted DATA is not compacted RESERVED
58	(3A)	SIGNED	2	DSNFSEQ	(REQUIRED) FILE SEQ NUMBER ON VOLUME
60	(3C)	CHARACTER	6	DSNVOL	(REQUIRED) VOLSER HOLDING DATA SET
66	(42)	SIGNED	2	DSNVOLSQ	(REQUIRED) VOLSEQ WITHIN SET OF VOLS
68	(44)	CHARACTER	4	DSNUNITA	UNIT WHERE DS WAS WRITTEN
72	(48)	CHARACTER	4	DSNRECFM	(REQUIRED) DS RECORD FORMAT
76	(4C)	SIGNED	4	DSNLRECL	(REQUIRED) DS LOGICAL RECORD LENGTH
80	(50)	SIGNED	4	DSNBLKSZ	(REQUIRED) DS BLOCK SIZE
84	(54)	SIGNED	4	DSNBLKCT	(REQUIRED) DS BLOCK COUNT
88	(58)	SIGNED	4	DSNKBYTS	(REQUIRED) DS SIZE IN KBYTES
92	(5C)	CHARACTER	8	DSNOWNER	(REQUIRED) DS OWNER NAME
100	(64)	SIGNED	2	DSNLFSEQ	(REQUIRED) LOGICAL FILE SEQ NUMBER
102	(66)	SIGNED	2	DSNSCLV	DATA SET SECURITY LEVEL
104	(68)	CHARACTER	4	DSLRDATE	(REQUIRED) DATE LAST READ YYYYDDD

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
108	(6C)	CHARACTER	4	DSLWDATE	(REQUIRED) DATE LAST WRITTEN YYYYDDD
112	(70)	CHARACTER	8	DSVRSVAL	VRS MANAGEMENT VALUE
120	(78)	CHARACTER	8	DSSMSMC	SMS MANAGEMENT CLASS
128	(80)	BITSTRING	1	DSFLAG	DSN FLAG BYTE
b		1...		DSFLGCAT	DATA SET IS CATALOGED
		.111		*	RESERVED
		.... 1...		DSFLGABD	ABEND IN PROCESS WHEN DATA SET
		.... .111		*	CLOSED
129	(81)	CHARACTER	3	*	RESERVED
132	(84)	CHARACTER	8	DDCRSID	CREATING SYSTEM ID
140	(8C)	CHARACTER	8	DSCJOBN	DATA SET CREATING JOBNAME
148	(94)	CHARACTER	8	DSNCSTEP	(COND) DSN1 Creating step name
156	(9C)	CHARACTER	8	DSNCDDNM	(COND) DSN1 Creating DD name
164	(A4)	CHARACTER	4	DSRETDTE	(COND) DATA SET RETENTION DATE
RESERVED OF					
168	(A8)	CHARACTER	8	DSNCPGM	CREATE PROGRAM NAME
176	(B0)	CHARACTER	8	DSNLPGM	LAST USED PROGRAM NAME
184	(B8)	CHARACTER	8	DSNLJOB	LAST USED JOB NAME
192	(C0)	CHARACTER	8	DSNLSTEP	LAST USED STEP NAME
200	(C8)	CHARACTER	8	DSNLDD	LAST USED DD NAME
208	(D0)	SIGNED	4	DSNTBLK	TOTAL BLOCK COUNT
212	(D4)		4	DSEXPDT	DATA SET EXPIRATION DATE
216	(D8)		4	DSEXPDT	ORIGINAL EXPIRATION DATE
220	(DC)	SIGNED	4	(5)	RESERVED
		1111		DRECEND	"*-DREC"
220	(DC)	1111		DRECEND	END OF CDREC

Archived

Constants

Len	Type	Value	Name	Description
1	CHARACTER	D	DSNRTYPE	‘D’ - DATASET RECORD TYPE

Cross reference

Name	Hex Offset	Hex Value	Level
CDREC	0		1
CDRECEND	C0		2
DDCRDATE	5		2
DDCRSID	84	40404040	2
DDCRTIME	9		2
DREC	0		2
DRECLEN	DC	F0	2
DRECRDW	0		2
DSCJOBN	84	40404040	2
DSEXPDT	D4	C	2
DSEXPDTO	D8	C	2
DSFLAG	80	0	2
DSFLGABD	80	08	3
DSFLGCAT	80	80	3
DSLRDATE	68	C	2
DSLWDATE	6C	C	2
DSNBLKCT	54	0	2
DSNBLKSZ	50	0	2
DSNCDDNM	9C	40404040	2
DSNCPGM	A8	40404040	2

Archived

Name	Hex Offset	Hex Value	Level
DSNCSTEP	94	40404040	2
DSNFSEQ	3A	1	2
DSNKBYTS	58	0	2
DSNLDD	C8	40404040	2
DSNLFSEQ	64	1	2
DSNLJOB	B8	40404040	2
DSNLPGM	B0	40404040	2
DSNLRECL	4C	0	2
DSNLSTEP	C0	40404040	2
DSNNAME	D	40404040	2
DSNOWNER	5C	40404040	2
DSNRECFM	48	40404040	2
DSNRID	4	C4	2
DSNRTYPE	4	C4	2
DSNSCLV	66	0	2
DSNTBLK	D0	0	2
DSNTIDRC	39	80	3
DSnTNCOM	39	40	3
DSNTNCOM	39	4	2
DSNTRTCH	39	0	2
DSNUNITA	44	40404040	2
DSNVOL	3C	40404040	2
DSNVOLSQ	42	1	2
DSRETDTE	A4	C	2
DSSMSMC	78	40404040	2
DSVRSVAL	70	40404040	2
RMMDREC		0	1

Archived

EDGCEREC record layout

```
          RMM CONVERSION EMPTY SHELF RECORD MAPPING
DFSMS/MVS 1.3.0 DFSMSrmm
PROPRIETARY V3 STATEMENT
LICENSED MATERIALS - PROPERTY OF IBM
"RESTRICTED MATERIALS OF IBM"
5695-DF1
(C) COPYRIGHT 1994,1995 IBM CORP.
END PROPRIETARY V3 STATEMENT
    FUNCTION : EDGCEREC MAPS THE CONVERSION EMPTY SHELF RECORD
    LOCATION : N/A
    METHOD OF ACCESS:
               ASSEMBLER - USING ON RMMEREC
               LA Rx,ERECAREA
               USING RMMEREC,Rx
               MVI ESRID,C'E' Set E record type
                   PLX - %INCLUDE EDGCEREC
LISTING : ASSEMBLER - SPECIFY LIST=YES ON THE MACRO
          PLX - SPECIFY &IHALIST='YES' BEFORE INCLUDE

CHANGE ACTIVITY:
$00=OW0xxxx,110,940920,MWW: Create Macro
$01=OW11151,110,940921,MWW: Implement Macro
$02=OW12935,120,951004,GD : Add supporting PLX/370
```

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
START OF RMMEREC					
0	(0)	STRUCTURE	68	CEREC	0
0	(0)	CHARACTER	EREC		BOUNDARY ALIGN
0	(0)	SIGNED	4	ERECRDW	RESERVED FOR RDW
4	(4)	CHARACTER	1	ESRID	(REQUIRED) EMPTY RACK RECORD ID
		11.. .1.1		ESRTYPE	'E' - EMPTY RACK RECORD TYPE
5	(5)	CHARACTER	1	ESTYPE	(REQUIRED) SHELF TYPE
		11.. ..1.		EBSHELF	'B' - BIN TYPE LOCATION RECORD
		11.1 1..1		ERSHELF	'R' - RACK TYPE SHELF RECORD
6	(6)	CHARACTER	8	ESLOCNAM	(REQUIRED) NAME OF ASSOCIATED LIBRY

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
14	(E)	CHARACTER	6	ESSTART	(REQUIRED) FIRST OF EMPTY SHELVES
20	(14)	CHARACTER	6	ESCOUNT	(REQUIRED) N-1 OF CONTIGUOUS SHELVES
26	(1A)	CHARACTER	8	EVMEDIA	(REQUIRED) MEDIA NAME
34	(22)	SIGNED	2	*	RESERVED
36	(24)	SIGNED	4	* (8)	RESERVED
68	(44)	CHARACTER		CNVEEND	END OF CNVEREC

Constants

Len	Type	Value	Name	Description
1	CHARACTER	E	ESRTYPE	‘E’ - EMPTY RACK RECORD
1	CHARACTER	R	ERSHELF	‘R’ - RACK TYPE SHELF
1	CHARACTER	B	EBSHELF	’B’ - BIN TYPE IN STORE

Cross reference

Name	Hex Offset	Hex Value	Level
CEREC	0		1
CNVEEND	44		2
EBSHELF	5	C2	2
EREC	0		2
ERECRDW	0		2
ERECRDW	0		2

Archived

Name	Hex Offset	Hex Value	Level
ERSHELF	5	D9	
ESCOUNT	14	40404040	2
ESLOCNAM	6	40404040	2
ESRID	4	C5	2
ESRTYPE	5	C5	2
ESSTART	E	40404040	2
ESTYPE	5	40	2
EVMEDIA	1A	40404040	2
RMMEREC	0		1

Archived

EDGCKREC record layout

```
          RMM CONVERSION VRS RECORD MAPPING
DFSMS/MVS 1.5.0 DFSMSrmm
PROPRIETARY V3 STATEMENT
LICENSED MATERIALS - PROPERTY OF IBM
"RESTRICTED MATERIALS OF IBM"
5695-DF1
(C) COPYRIGHT 1994,1995 IBM CORP.
END PROPRIETARY V3 STATEMENT
    FUNCTION : EDGCKREC MAPS THE CONVERSION VRS RECORD
    LOCATION : N/A
    METHOD OF ACCESS:
                ASSEMBLER - USING ON RMMKREC
                LA Rx,KRECAREA
                USING RMMKREC,Rx
                MVI VRSRID,C'K' Set K record type
                    PLX - %INCLUDE EDGCKREC
LISTING : ASSEMBLER - SPECIFY LIST=YES ON THE MACRO
          PLX - SPECIFY &IHALIST='YES' BEFORE INCLUDE

CHANGE ACTIVITY:
$00=OW0xxxx,110,940920,MWW : Create Macro
$01=OW11151,110,940921,MWW : Implement Macro
$02=OW12935,120,951004,GD  : Add supporting PLX/370
$K1=K151038,150,980616,BDG : Correct flags in PLX/370 part
$S7=OW30969,140,971001,RS : VRS Enhancements
$03=OW32993,130,980607,RS  : sync PL/X and Assembler part
```

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
START OF RMMKREC					
0	(0)	STRUCTURE	184	CKREC	0
0	0	CHARACTER	0	KREC	BOUNDARY ALIGN
0	(0)	SIGNED	4	KRECRDW	RESERVED FOR RDW
4	(4)	CHARACTER 11.1 ..1.	1	VRSRID VRSRTYPE	(REQUIRED) VRS RECORD ID 'K' - VITAL RECORDS RECORD TYPE
5	(5)	CHARACTER	4	VCRDATE	VRS CREATE DATE - YYYYDDD
9	(9)	CHARACTER	4	VCRTIME	VRS CREATE TIME - HHMMSST

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
13	(D)	CHARACTER	1	VTYPEID	(REQUIRED) VRS TYPE
		111. .1..		VTYPED	'D' - DATASET TYPE VRS
		111. .1.1		VTYPEN	'N' - NAME TYPE VRS
		111. .1.1		VTYPEV	'V' - VOLUME TYPE VRS
14	(E)	CHARACTER	44	VDSNAME	(COND) DATASET NAME
58	(3A)	BITSTRING	1	VDSNTYPE	(COND) FORMAT OF DATA SET NAME
		1...		VDSNGDG	GDG FORMAT DSN
		.1..		*	
		..1.		VDSNSTD	STD FORMAT DSN
		...1 1111		*	
59	(3B)	BITSTRING	1	VRETN	VRS RETENTION TYPE
		1...		VRETNCYC	RETAIN® BY CYCLES
		.1..		VRETNELP	RETAIN BY ELAPSED DAYS
		..1.		VRETNLRF	RETAIN BY DAYS SINCE REF
		...1		VRETNCAT	RETAIN WHILST CATALOGUED
		.... 1...		VRETNUEX	RETAIN UNTIL EXPIERED
		.... .1..		VRETNXD	EXTRADAYS
		.... ..1.		VRETNBDC	BYDAYSCYCLE
		1		*	
60	(3C)	CHARACTER	6	VVOLSER	(COND) VOLUME SERIAL NUMBER
66	(42)	SIGNED	2	VDELAY	(REQUIRED) DAYS DELAY BEFORE MOVEMNT
68	(44)	SIGNED	4	VRCOUNT	(REQUIRED) VITAL RECORDS NUMBER. NUMBER OF CYCLES OR DAYS FOR VOLUME TO BE KEPT
72	(48)	CHARACTER	8	VRSNAME	(COND) VRS NAME
80	(50)	SIGNED	4	VSTRKEEP	(REQUIRED) STORE KEEP NUMBER. NUMBER OF CYCLES OR DAYS OF VOLUME TO BE KEPT IN STORE
84	(54)	CHARACTER	8	VOWNER	(REQUIRED) VRS OWNER ID
92	(5C)	CHARACTER	4	VDELDATE	AUTO-DELETE DATE -YYYYDDD
96	(60)	CHARACTER	30	VDESC	VRS DESCRIPTION
126	(7E)	CHARACTER	8	VLOCNAM	(REQUIRED) LOCATION NAME
134	(86)	CHARACTER	8	VRSNEXT	(COND) NAME OF CHAINED VRS

Archive

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
142	(8E)	CHARACTER	8	VRSJOBN	(COND) NAME OF CREATING JOB
150	(96)	BITSTRING	1	VRSOPTS	OPTIONS BYTE
		1...		VANDVRS	VRSNEXT IS ANDVRS()
		.1..		VRELIXD	EXPIRY DATE IGNORE
		..1.		VRELSI	SCRATCH IMMEDIATE
		...1 1111		*	RESERVED
151	(97)	CHARACTER	1	*	RESERVED
152	(98)	SIGNED	4	* (8)	RESERVED
184	(B8)	CHARACTER		CKRECEND	END OF CNVKREC

Constants

Len	Type	Value	Name	Description
1	CHARACTER	K	VRSTYPE	'K' - VRS RECORD TYPE
1	CHARACTER	D	VTYPED	'D' - DATA SET TYPE VRS
1	CHARACTER	N	VTYPEN	'N' - NAME TYPE VRS
1	CHARACTER	V	VTYPEV	'V' - VOLUME TYPE VRS

Cross reference

Name	Hex Offset	Hex Value	Level
CKREC	0		1
CKRECEND	B8		2
KREC	0		2
KRECLEN	98	B8	2
KRECRDW	0		2
RMMKREC	0		1
VANDVRS	96	80	2

Archived

Name	Hex Offset	Hex Value	Level
VCRDATE	5	C	2
VCRTIME	9	C	2
VDELAY	42	0	2
VDELDATE	5C	o	2
VDESC	60	40404040	2
VDSNAME	E	40404040	2
VDSNGDG	3A	80	3
VDSNSTD	3A	20	3
VDSNTYPE	3A	20	2
VLOCNAM	7E	C8D6D4C5	2
VOWNER	54	40404040	2
VRCOUNT	44	1869F	2
VRELIXD	96	40	3
VRELSI	96	20	3
VRETN	3B	0	2
VRETNBDC	3B	02	3
VRETNCAT	3B	10	3
VRETNCYC	3B	80	3
VRETNELP	3B	40	3
VRETNLRF	3B	20	3
VRETNUEX	3B	08	3
VRETNXD	3B	04	3
VRSJOBN	8E	40404040	2
VRSNAME	48	40404040	2
VRSNEXT	86	40404040	2
VRSOPTS	96	0	2
VRSRID	4	D2	2
VRSTYPE	4	D2	2
VSTRKEEP	50	1869F	2

Archived

Name	Hex Offset	Hex Value	Level
VTYPEC	D	E5	2
VTYPED	D	C4	2
VTYPEID	D	C4	2
VTYPEN	D	D5	2
VVOLSER	3C		2

Archived

EDGCLREC record layout

```
          RMM CONVERSION LIBRARY RECORD MAPPING
z/OS DFSMSrmm V1R3
PROPRIETARY V3 STATEMENT
LICENSED MATERIALS - PROPERTY OF IBM
"RESTRICTED MATERIALS OF IBM"
5694-A01
(C) COPYRIGHT 1993,2003 IBM CORP.
STATUS=HDZ11G0
END PROPRIETARY V3 STATEMENT
    FUNCTION : EDGCLREC MAPS THE CONVERSION LIBRARY RECORD
    LOCATION : N/A
    METHOD OF ACCESS:
               ASSEMBLER - USING ON RMMLREC
               LA Rx,LRECAREA
               USING RMMLREC,Rx
               MVI LIBRID,C'L' Set L record type
                   PLX - %INCLUDE EDGCLREC
LISTING : ASSEMBLER - SPECIFY LIST=YES ON THE MACRO
          PLX - SPECIFY &IHALIST='YES' BEFORE INCLUDE

CHANGE ACTIVITY:
$00=OW0xxxx,110,940920,MWW: Create Macro
$01=OW11151,110,940921,MWW: Implement Macro
$02=OW12935,120,951004,GD : Add supporting PLX/370
$03=OW18740,120,960227,WKN: Mismatch between PLX and ASSM
$04=OW21275,120,960916,CHK: Alignment
$K1=K151038,150,980616,BDG: Correct offsets in ASM part
$05=OW32993,130,980607,RS : sync PL/X and Assembler part
$06=OW37600,140,990112,RS : Add EHPCT media and SMS management class support
$07=OW40710,140,990819,RS : Add Creating program name
$08=OW44589,140,000518,OG : Add NO DATA SET RECORDING flag
$LL=RMM213 ,213,010216,BDG: >9999 tape files
$09=OW48290,160,010220,RS : Add 256 track media support
$10=OW51797,210,011002,MWW: Add Use Count
$LS=RMMV1R5,1R5,020122,BG : Duplicate volser support
$11=OW56705,210,021004,MWW: BLKSIZE incorrect after conversion
$12=OW57054,210,021104,MWW: Add 384 track media support
$13=OA02095,213,030224,AK : Data set expiration date support
$14=OA02206,              : Support 3592
```

Abbreviated

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
START OF RMMLREC					
0	(0)	STRUCTURE	580	CLREC	0
0	(0)	CHARACTER		LREC	BOUNDARY ALIGN
0	(0)	SIGNED	4	LRECRDW	RECORD DESCRIPTOR WORD
4	(4)	CHARACTER 11.1 ..11	1	LIBRID LVTYPE	(REQUIRED) LIBRARY RECORD ID ' L' - LIBRARY RECORD TYPE
5	(5)	CHARACTER	6	LVOLSER	(REQUIRED) VOLUME SERIAL NUMBER
11	(B)	CHARACTER	8	LVOWNER	(REQUIRED) VOLUME OWNER
19	(13)	CHARACTER	8	LVOUID	VOLUME NOTIFY USERID
27	(1B)	CHARACTER	8	LVONODE	VOLUME NOTIFY NODE
35	(23)	CHARACTER	1	LRESRVA	RESERVED
36	(24)	SIGNED	2	LVSECLEV	VOLUME SECURITY LEVEL
38	(26)	SIGNED	2	LVSEQNO	(REQUIRED) VOLUME SEQUENCE NUMBER
40	(28)	CHARACTER	4	LVCRDATE	(REQUIRED) VOLUME CREATE DATE -YYYYDDD
44	(2C)	CHARACTER	4	LVCRTIME	VOLUME CREATE TIME - HHMMSST
48	(30)	CHARACTER	8	LVCRUID	VOLUME CREATE USERID
56	(38)	CHARACTER	8	LVCRJOB	VOLUME CREATE JOBNAME
64	(40)	BITSTRING	4	LVUCBTYP	CUPY OF UCBTYP
68	(44)	BITSTRING	4	LVTDSI	(COND) TAPE MEDIA TYPE INFORMATION
68	(44)	BITSTRING 1111 11.. ....1 .... ..1. .... ..11 .... .1.. .... .1.1 .... .11.	1	LVTDSRC * LVM18TRK LVM36TRK LVM128TR LVM256TR LVM384TR LVMEFMT1	TAPE RECORDING FORMAT. ONE OF: X'00' - NOT 3480 X'01' - 18 TRACK X'02' - 36 TRACK X'03' - 128 TRACK X'04' - 256 TRACK X'05' - 384 TRACK X'06' - EFMT1 (3592)

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
69	(45)	BITSTRING	1	LVTDSTY	TAPE MEDIA TYPE. ONE OF:
		1111 11..		*	X'00' - NOT 3480
		1		LVMSTDCP	X'01' - STANDARD CAPACITY
		.... ..1.		LVMENHCP	X'02' - ENHANCED CAPACITY
		11		LVMHPCT	X'03' - HPCT 3590 STD CAP
		.... .1..		LVMEHPCTLLVM	X'04' - EHPCT 3590 EXT CAP
		.... .1.1		LVMMED5	X'05' - MEDIA5 for 3592
70	(46)	BITSTRING	1	LVTDSCM	TAPE COMPACTION TECHNIQUE. ONE OF:
		1111 11..		*	
		1		LVMNCOMP	X'00' - UNKNOWN
		.... ..1.		LVMIDRC	X'01' - NOT COMPACT X'02' - IDRC
71	(47)	BITSTRING	1	LVTDSSA	TAPE SPECIAL ATTRIBUTES. ONE OF:
		1111 111.		*	X'00' - NONE
		1		LVM18TRD	X'01' - 18 TRACK READ ONL
72	(48)	SIGNED	4	LVTUSE	TAPE USAGE IN K-BYTES
76	(4C)	CHARACTER	4	LVEXDATO	(REQUIRED) JCL VOL EXPIRATION DATE
80	(50)	CHARACTER	4	LVEXDATE	(REQUIRED) VOLUME EXPIRATION DATE
84	(54)	CHARACTER	4	LVRETDTE	(COND) VOLUME RETENTION DATE
88	(58)	SIGNED	2	LVDSNCNT	(REQUIRED) COUNT OF DS W/ORIG ONVOL
90	(5A)	SIGNED	2	LVTRERR	TEMPORARY READ ERRORS
92	(5C)	SIGNED	2	LVTWERR	TEMPORARY WRITE ERRORS
94	(5E)	SIGNED	2	LVPRERR	PERMANENT READ ERRORS
96	(60)	SIGNED	2	LVPWERR	PERMANENT WRITE ERRORS
98	(62)	BITSTRING	1	LVRDEN	COPY OF JFCBDEN
99	(63)	CHARACTER	1	LBPIIND	(REQUIRED) BPI INDICATOR
		1111 ..11		BPI1600	'3' - 1600BPI
		1111 .1..		BPI6250	'4' - 6250BPI
		1111 1..1		BPI3480	'9' - 3480
		11.. ..11		BPI3480C	'C' - 3480 COMPACT (IDRC)
		.1.1. 11..		BPIUNDEF	' *' - UNDEFINED

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
100	(64)	BITSTRING	1	LVFLAGA	(REQUIRED) FLAGS 'A' - STATUS
		1...		LVMSTFLG	VOLUME IS MASTER
		.1..		LVRLSFLG	VOLUME PENDING RELEASE
		..1.		LVVRFLG	VITAL RECORD - DO NOT RELEASE
		...1		LVASSFLG	USER TAPE (ASSIGNED BY LIB)
		.... 1...		LVLONFLG	TAPE IS ON LOAN
		.... .1..		LVOPNFLG	TAPE OPENED AND NOT YET CLOSED
		.... ..1.		LVSCRFLG	VOLUME IS SCRATCH
		1		LVOCEFLG	VOLUME RECORDED BY O/C/EOV
101	(65)	BITSTRING	1	LVFLGAX	(REQUIRED) FLAGS ' A' - STATUS
		1...		LVGVCFLG	EXTENSION
		.1..		LVXINFLG	SCRATCH VOL CLAIMED VIA GETVOL
		..1.		LVINIFLG	SCRATCH VOL NEVER INITIALIZED
		...1		LVENTFLG	SCRATCH VOL W/INIT ACTION PEND
		.... 1...		LVFABEND	SCRATCH VOL WAIT TO ENTER ATL
		.... .1..		LVFOCEAB	ABEND IN PROCESS WHEN DS CLOSED
					ABEND PROBABLY IN O/C/EOV
102	(66)	BITSTRING	1	LVFLAGB	(REQUIRED) FLAGS ARE 0 OR 1
		1...		LVDEFRET	DEFAULT. RETEN. PERIOD USED
		.1..		*	
		..1.		LVNLTAPE	LABEL TYPE IS NL
		...1		LVALTAPE	LABEL TYPE IS AL
		.... 1...		LVSLTAPE	LABEL TYPE IS SL
		.... .1..		*	
		.... ..1.		LVBLTAPE	TAPE LAST WRITTEN USING BLP
		1		LVULTAPE	SL OR AL W/USER LABELS
103	(67)	BITSTRING	1	LVFLAGC	(REQUIRED) FLAGS ' C' - RELEASE
		1...		LVRETSCR	ACTIONS
		.1..		LVREPREL	RETRN TO SCRTCH POOL - DEFLT
		..1.		LVREINIT	REPLACE TAPE ON RELEASE
		...1		LVDEGAUS	REINITIALISE
		.... 1...		LVROWNER	DEGAUS/SECURITY ERASE
		.... .1..		LVNOWNER	RETURN TO OWNER
					NOTIFY OWNER

Archive

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
104	(68)	BITSTRING	1	LVFLAGD	(REQUIRED) FLAGS ' D' - ACCESS
		1...		LVOREAD	OWNER MAY READ VOLUME
		.1..		LVOUPD	OWNER MAY UPDATE VOLUME
		..1.		LVOALT	OWNER MAY ALTER VOLUME
		...1		LVPROTR	READ-ONLY PROTECTION
		.... 1...		LVPROTU	UPDATE PROTECTION
		.... .1..		LVMVSUSE	MAY BE USED ON MVS SYSTEMS
		.... ..1.		LVVMUSE	MAY BE USED ON VM SYSTEMS
		1		LVNODSNR	NO DATA SET RECORDING
105	(69)	CHARACTER	8	LVMEDIA	MEDIA NAME
113	(71)	CHARACTER	4	LVUNITAD	UNIT ADDRESS USED
117	(75)	CHARACTER	4	LVLSTUAD	LAST UNIT ADDRESS USED
121	(79)	CHARACTER	8	LVLONLOC	LOAN LOCATION
129	(81)	CHARACTER	4	LVLRDDAT	DATE VOLUME LAST READ (YYYYDDD)
133	(85)	CHARACTER	4	LVLWTDAT	DATE VOLUME LAST WRIT (YYYYDDD)
137	(89)	CHARACTER	4	LVASDATE	ASSIGN FROM SCR DATE (YYYYDDD)
141	(8D)	CHARACTER	4	LVASTIME	ASSIGN FROM SCR TIME (HHMMSST)
145	(91)	BITSTRING	1	LVLTYP	COPY OF JFCBLTYP
146	(92)	BITSTRING	1	LVTRTCH	(REQUIRED) FROM JFCTRTCH - IDRC
		1111		*	SUPPORT
		.... 1...		LVTCOMP	
		.... .1..		LVTNCOMP	DSN USED 3480 IDRC NO COMPACTION
147	(93)	CHARACTER	1	LRESRVB	RESERVED
148	(94)	CHARACTER	8	LVLOCNAM	(REQUIRED) CURRENT LOCATION OF VOLUME
156	(9C)	CHARACTER	8	LVDEST	DESTINATION NAME
164	(A4)	CHARACTER	8	LVHLOC	HOME LOCATION NAME
172	(AC)	CHARACTER	8	LVSGNAME	STORAGE GROUP NAME
180	(B4)	CHARACTER	40	LACCINF	ACCOUNTING INFORMATION
220	(DC)	CHARACTER	30	LDESC	USER DESCRIPTION

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
250	(FA)	CHARACTER	96	LAUTHIDS	AUTHORISED USER IDS
250	(FA)	CHARACTER	8	LAUTHID1	1ST USERID SLOT
258	(102)	CHARACTER	8	LAUTHID2	2ND USERID SLOT
266	(10A)	CHARACTER	8	LAUTHID3	3RD USERID SLOT
274	(112)	CHARACTER	8	LAUTHID4	4TH USERID SLOT
282	(11A)	CHARACTER	8	LAUTHID5	5TH USERID SLOT
290	(122)	CHARACTER	8	LAUTHID6	6TH USERID SLOT
298	(12A)	CHARACTER	8	LAUTHID7	7TH USERID SLOT
306	(132)	CHARACTER	8	LAUTHID8	8TH USERID SLOT
314	(13A)	CHARACTER	8	LAUTHID9	9TH USERID SLOT
322	(142)	CHARACTER	8	LAUTHIDA	10TH USERID SLOT
330	(14A)	CHARACTER	8	LAUTHIDB	11TH USERID SLOT
338	(152)	CHARACTER	8	LAUTHIDC	12TH USERID SLOT
346	(15A)	CHARACTER	6	LRACKNO	(REQUIRED) RACK NUMBER
352	(160)	CHARACTER	6	LRSBIN	(COND) REMOTE STORE BIN NUMBER
358	(166)	CHARACTER	4	LRSDATE	(COND) REMOTE STORE DATE (YYYYDDD)
362	(16A)	CHARACTER	8	LRBINMDN	(COND) BIN MEDIA NAME
370	(172)	SIGNED	2	LLABNO1	(REQUIRED) LABEL NUMBER OF FIRST FILE
372	(174)	CHARACTER	44	LDS1NAM	NAME OF 1ST DATASET ON VOLUME
416	(1A0)	CHARACTER	8	LDS1OWNR	DATASET OWNER
424	(1A8)	SIGNED	2	LDS1SEQ	(COND) SEQ NO OF DATA SET ONE
426	(1AA)	CHARACTER	4	LDS1RECF	(COND) FIRST DATA SET' S RECFM
430	(1AE)	CHARACTER	2	*	ALIGNMENT
432	(1B0)	SIGNED	4	LDS1LREC	(COND) FIRST DATA SET'S LRECL
436	(1B4)	SIGNED	4	LDS1BSIZ	(COND) FIRST DATA SET'S BLKSZ

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
440	(1B8)	SIGNED	4	LDS1BLKS	(COND) FIRST DATA SET'S BLK COUNT
444	(1BC)	SIGNED	4	LDS1KBYT	(COND) FIRST DATA SET'S K-BYTES
448	(1C0)	SIGNED	2	LDS1SCLV	FIRST DATA SET SECURITY LEVEL
450	(1C2)	CHARACTER	4	LDCRDATE	(REQUIRED) DSN1 CREATE DATE YYYYDDD
454	(1C6)	CHARACTER	4	LDCRTIME	DSN1 CREATE TIME - HHMMSST
458	(1CA)	CHARACTER	4	LDLRDATE	(COND) DSN1 DATE LAST READ - YYYYDDD
462	(1CE)	CHARACTER	4	LDLWDATE	(COND) DSN1 DATE LAST WRIT - YYYYDDD
466	(1D2)	BITSTRING 1... .... .111 .... 1...	1	LDS1FLAG LDS1FCAT * LDS1FABD	DSN1 FLAG BYTE X'80' - DATA SET IS CATALOGED RESERVED X'08' - ABEND IN PROCESS WHEN DS CLOS
467	(1D3)	CHARACTER	8	LDVRSVAL	(COND) VRS MANAGEMENT VALUE
475	(1DB)	CHARACTER	8	LDSMSMC	(COND) SMS MANAGEMENT CLASS
483	(1E3)	CHARACTER	8	LDS1STEP	(COND) DSN1 CREATING STEP NAME
491	(1EB)	CHARACTER	8	LDS1DDNM	(COND) DSN1 CREATING DD NAME
499	(1F3)	CHARACTER	4	LDRETDTE	(COND) DATA SET RETENTION DATE
503	(1F7)	CHARACTER	44	LDSXNAME	(COND) NAME OF LAST DATASET ON VOL
547	(223)	CHARACTER	8	LVCRSID	CREATING SYSTEM ID
555	(22B)	CHARACTER	8	LDS1CPGM	CREATE PROGRAM NAME
563	(233)	CHARACTER	8	LDS1LPGM	LAST USED PROGRAM NAME
571	(23B)	CHARACTER	8	LDS1LJOB	LAST USED JOB NAME
579	(243)	CHARACTER	8	LDS1LSTP	LAST USED STEP NAME
587	(24B)	CHARACTER	8	LDS1LDD	LAST USED DD NAME
595	(253)	CHARACTER	1		RESERVED
596	(254)	CHARACTER	2	LVUSE	VOLUME USE COUNT

Archived

Offsets					
Dec	Hex	Type	Len	Name (Dim)	Description
598	(256)	CHARACTER	6	LVVOL1	(REQUIRED) VOL1 VOLSER NUMBER
604	(25C)	CHARACTER	4	LDS1TBLK	TOTAL BLOCK COUNT
608	(260)		4	LDS1EXP	DSN1 EXPIRATION DATE
612	264		4	LDS1EXPO	DSN1 ORIGINAL EXPIRATION DATE
616	268	SIGNED		(3)	RESERVED
616	(274)	CHARACTER		CLRECEND	END OF CNVLREC

Constants

Len	Type	Value	Name	Description
1	CHARACTER	L	LVTYPE	'L' - LIBRARY RECORD TYPE
1	CHARACTER	3	BPI1600	'3' - 1600BPI
1	CHARACTER	4	BPI6250	'4' - 6250BPI
1	CHARACTER	9	BPI3480	'9' - 3480
1	CHARACTER	C	BPI348C	'C' - 3480 COMPACT (IDRC)
1	CHARACTER	*	BPIUNDEF	'*' - UNDEFINED

Cross reference

Name	Hex Offset	Hex Value	Level
ALTAPE	66	10	2
ASSFLG	64	10	2
BLTAPE	66	2	2
BPI1600	63	F3	2
BPI3480	63	F9	2
BPI3480C	63	C3	2

Archived

Name	Hex Offset	Hex Value	Level
BPI6250	63	F4	2
BPIUNDEF	63	5C	2
CLREC	0		1
CLRECEND	270		2
COMPACT	92	8	2
DEFRET	66	80	2
DEGAUS	67	10	2
LACCINF	B4	40404040	2
LAUTHID1	FA		3
LAUTHID2	102		3
LAUTHID3	10A		3
LAUTHID4	112		3
LAUTHID5	11A		3
LAUTHID6	122		3
LAUTHID7	12A		3
LAUTHID8	132		3
LAUTHID9	13A		3
LAUTHIDA	142		3
LAUTHIDB	14A		3
LAUTHIDC	152		3
LAUTHIDS	FA	40404040	2
LBPIIND	63		2
LDCRDATE	1C2	C	2
LDCRTIME	1C6	C	2
LDESC	DC	40404040	2
LDLRDATE	1CA	C	2
LDLWDATE	1CE	C	2

Archived

Name	Hex Offset	Hex Value	Level
LDRETDTE	1F3	C	2
LDS1BLKS	1B8	0	2
LDS1BSIZ	1B4	0	2
LDS1CPGM	243	40404040	2
LDS1DDNM	1EB	40404040	2
LDS1EXP	260	C	2
LDS1EXPO	264	C	2
LDS1FABD	1D2	08	3
LDS1FCAT	1D2	80	3
LDS1FLAG	1D2	0	2
LDS1KBYT	1BC	0	2
LDS1LDD	263	40404040	2
LDS1LJOB	253	40404040	2
LDS1LPGM	24B	40404040	2
LDS1LREC	1B0	0	2
LDS1LSTP	25B	40404040	2
LDS1NAM	174	40404040	2
LDS1OWNR	1A0	40404040	2
LDS1RECF	1AA	40404040	2
LDS1SCLV	1C0	0	2
LDS1SEQ	1A8		2
LDS1STEP	1E3	40404040	2
LDS1TBLK	258	0	2
LDSMSMC	1DB	40404040	2
LDSMSMC	1DB	40404040	2
LDSXNAME	1F7	40404040	2
LDSXNAME	1F7		2

Archived

Name	Hex Offset	Hex Value	Level
LDVRSVAL	1D3	40404040	2
LIBRID	4	D3	2
LLABNO1	172	1	2
LONFLG	64	8	2
LRACKNO	15A	40404040	2
LRBINMDN	16A	40404040	2
LREC	0		2
LRECLEN	268	274	2
LRECRDW	0		2
LRESRVA	23	0	2
LRESRVB	93	40	2
LRSBIN	160	40404040	2
LRSDATE	166	C	2
LVASDATE	89	C	2
LVASTIME	8D	C	2
LVBLTAPE	66	02	3
LVCRDATE	28	C	2
LVCRJOB	38	40404040	2
LVCRSID	23B	40404040	2
LVCRTIME	2C	C	2
LVCRUID	30	40404040	2
LVDEST	9C	40404040	2
LVDSNCNT	58	0	2
LVEXDATE	50	C	2
LVEXDATO	4C	C	2
LVFLAGA	64		2
LVFLAGB	66		2

Archived

Name	Hex Offset	Hex Value	Level
LVFLAGC	67	0	2
LVFLAGD	68		2
LVFLGAX	65		2
LVHLOC	A4	40404040	2
LVLOCNAM	94	40404040	2
LVLONLOC	79	40404040	2
LVLRDDAT	81	C	2
LVLSTUAD	75	40404040	2
LVLTYP	91	0	2
LVLWTDAT	85	C	2
LVM128TR	44	03	3
LVM18TRK	44	01	3
LVM256TR	44	04	3
LVM36TRK	44	02	3
LVM384TR	44	05	3
LVMEDIA	69	40404040	2
LVMEFMT1	44	06	3
LVMEHPCT	45	04	4
LVMENHCP	45	02	4
LVMHPCT	45	03	4
LVMIDRC	46	02	4
LVMMED5	45	05	3
LVMNCOMP	46	01	4
LVMSTDCP	45	01	4
LVMVSUSE	68	04	3
LVNLTAPE	66	20	3
LVNOWNER	67	04	3

Archived

Name	Hex Offset	Hex Value	Level
LVOALT	68	20	3
LVOCEFLG	64	01	3
LVOLSER	5	40404040	2
LVONODE	1B	40404040	2
LVOPNFLG	64	04	3
LVOREAD	68	80	3
LVOUID	13	40404040	2
LVOUPD	68	40	3
LVOWNER	B	40404040	2
LVPRERR	5E	0	2
LVPROTR	68	10	3
LVPROTU	68	08	3
LVPWERR	60	0	2
LVRDEN	62		2
LVREINIT	67	20	3
LVREPREL	67	40	3
LVRETDTE	54	C	2
LVRETSCR	67	80	3
LVRLSFLG	64	40	3
LVROWNER	67	08	3
LVSCRFLG	64	02	3
LVSECLEV	24	0	2
LVSEQNO	26		2
LVSGNAME	AC	40404040	2
LVSLTAPE	66	08	3
LVTCOMP	92	08	3
LVTDSCM	46		3

Archived

Name	Hex Offset	Hex Value	Level
LVTDSI	44	0	2
LVTDSRC	44		3
LVTDSSA	47		3
LVTDSTY	45		3
LVTNCOMP	92	04	3
LVTRERR	5A	0	2
LVTRTCH	92	0	2
LVTUSE	48	0	2
LVTWERR	5C	0	2
LVTYPE	4	D3	2
LVUCBTYP	40	0	2
LVULTAPE	66	01	3
LVUNITAD	71	40404040	
LVUSE	254		2
LVVMUSE	68	02	3
LVVOL1	256	40404040	2
LVVRFLG	64	20	3
LVXINFLG	65	40	3
MSTFLG	64	80	2
MVENTFLG	65	10	2
MVFABEND	65	08	2
MVFOCEAB	65	04	2
MVGVCFLG	65	80	2
MVINIFLG	65	20	2
MVSUSE	68	04	2
MVXINFLG	65	40	2
NCOMPACT	92	04	2

Archived

Name	Hex Offset	Hex Value	Level
NLTAPE	66	20	2
NODSNR	68	01	2
NOWNER	67	04	2
OALT	68	20	2
OCEFLG	64	01	2
OPNFLG	64	04	2
OREAD	68	80	2
OUPD	68	40	2
REINIT	67	20	2
REPREL	67	20	2
RETSCR	67	80	2
RLSFLG	64	40	2
RMMLREC	0		1
ROWNER	67	08	2
RPROT	68	10	2
SCRFLG	64	02	2
SLTAPE	66	08	2
ULTAPE	66	01	2
UPROT	68	08	2
VMUSE	68	02	2
VRFLG	64	20	2

Archived

EDGCOREC record layout

```
          RMM CONVERSION OWNER RECORD MAPPING
DFSMS/MVS 1.3.0 DFSMSrmm
PROPRIETARY V3 STATEMENT
LICENSED MATERIALS - PROPERTY OF IBM
"RESTRICTED MATERIALS OF IBM"
5695-DF1
(C) COPYRIGHT 1994,1995 IBM CORP.
END PROPRIETARY V3 STATEMENT
    FUNCTION : EDGCOREC MAPS THE CONVERSION OWNER RECORD
    LOCATION : N/A
    METHOD OF ACCESS:
                ASSEMBLER - USING ON RMMOREC
                LA Rx,ORECAREA
                USING RMMOREC,Rx
                MVI OWNERRID,C'O' Set O record type
               PLX - %INCLUDE EDGCOREC
LISTING : ASSEMBLER - SPECIFY LIST=YES ON THE MACRO
          PLX - SPECIFY &IHALIST='YES' BEFORE INCLUDE

CHANGE ACTIVITY:
$00=OW0xxxx,110,941011,MWW: Create Macro
$01=OW11151,110,941011,MWW: Implement macro
$02=OW12935,120,951004,GD: Supporting PLX/370
```

Offsets

Dec	Hex	Type	Len	Name (Dim)	Description
START OF RMMOREC					
0	(0)	STRUCTURE	292	COREC	0
0	(0)	CHARACTER		OREC	COREC INFORMATION
0	(0)	SIGNED	4	ORECRDW	RDW
4	(4)	CHARACTER	1	OWNERRID	OWNER' S NAME RECORD ID
		11.1 .11.		OWNRTYPE	'O' - OWNER RECORD TYPE
5	(5)	CHARACTER	8	OWNER	OWNER'S NAME
13	(D)	CHARACTER	20	OWNSNAME	OWNER'S SURNAME
33	(21)	CHARACTER	20	OWNFNAME	OWNER'S FIRST NAME

Archived

Offsets					
Dec	**Hex**	**Type**	**Len**	**Name (Dim)**	**Description**
53	(35)	CHARACTER	40	OWNDEPT	*(REQUIRED)* OWNER'S DEPARTMENT
93	(5D)	CHARACTER	40	OWNADDR1	OWNER'S 1ST ADDRESS LINE
133	(85)	CHARACTER	40	OWNADDR2	OWNER'S 2ND ADDRESS LINE
173	(AD)	CHARACTER	40	OWNADDR3	OWNER'S 3RD ADDRESS LINE
213	(D5)	CHARACTER	8	OWNINTEL	OWNER'S INTERNAL TELE NO.
221	(DD)	CHARACTER	20	OWNEXTEL	OWNER'S EXTERNAL TELE NO.
241	(F1)	CHARACTER	8	OWNUID	OWNER'S USERID
249	(F9)	CHARACTER	8	OWNNODE	OWNER'S NODE NAME
257	(101)	CHARACTER	3	OWNPAD1	PADDING
260	(104)	SIGNED	4	* (8)	RESERVED
292	(124)	CHARACTER		CNVOEND	END OF CNVOREC

Constants

Len	Type	Value	Name	Description
1	CHARACTER	O	OWNRTYPE	'O' - OWNER RECORD TYPE

Cross reference

Name	Hex Offset	Hex Value	Level
CNVOEND	124		2
COREC	0		1
OREC	0		2
ORECLEN	104	114	
ORECRDW	0		2
OWNADDR1	5D	40404040	2

Archived

Name	Hex Offset	Hex Value	Level
OWNADDR2	85	40404040	2
OWNADDR3	AD	40404040	2
OWNDEPT	35	40404040	2
OWNER	5	40404040	2
OWNERRID	4	D6	2
OWNEXTEL	DD	40404040	2
OWNFNAME	21	40404040	2
OWNINTEL	D5	40404040	2
OWNNODE	F9	40404040	2
OWNPAD1	101	404040	2
OWNRTYPE	4	D6	2
OWNSNAME	D	40404040	2
OWNUID	F1	40404040	2
RMMOREC	0		1

Archived

Archived

C

Security topics

This appendix provides you with the information you need to implement your installation security policies in DFSMSrmm using the FACILITY class profiles, and to activate RACF TAPEVOL and TAPEDSN classes. To protect DFSMSrmm functions, you need to use an external security product, such as RACF. You can use the information provided here, and the information provided in the *DFSMSrmm Implementation and Customization Guide*, SC26-7405, and *z/OS Security Server RACF Security Administrator's Guide*, SA22-7683 for authorizing users and ensuring security.

While the focus is on RACF, other security products will be covered where possible.

© Copyright IBM Corp. 2004. All rights reserved.

C.1 RACF implementation

In this section we take you through the process of using RACF to implement security for DFSMSrmm resources and functions in your installation.

The following list shows all the activities that should be done before you can implement DFSMSrmm:

- Assign a RACF user ID to DFSMSrmm
- Identify DFSMSrmm to RACF
- Define DFSMSrmm resources to RACF
- Define RACF groups for DFSMSrmm users
- Define DFSMSrmm resources to RACF
- Define the STGADMIN.ADR.DUMP.CNCURRNT resource

Note: You have to define the STGADMIN.ADR.DUMP.CNCURRNT resource only when you use DFSMSdss with concurrent copy.

C.1.1 Assign a RACF user ID for DFSMSrmm

For a started procedure to access any of the system resources, a user ID must be associated with that task. The RACF user ID for DFSMSrmm does not need to match the name of the procedure name you created in PROCLIB; any installation-selected RACF user ID is acceptable.

The user ID should have all the proper authorizations for accessing the system resources. We recommend defining a new user ID for each started procedure rather than using the default started task user ID (for example, STCUSER). In our system we associated user ID *DFRMM* with the DFSMSrmm started procedure.

All data sets are created for use by the DFSMSrmm procedure, add the DFSMSrmm RACF user ID to the access list for the data sets. Figure C-1 shows the RACF command we used to define the DFSMSrmm user ID.

```
ADDUSER DFRMM DFLTGRP(SYS1) NAME('DFSMSrmm Userid')
```

Figure C-1 RACF command ADDUSER

Note: If you do not specify the DFLTGRP parameter on the ADDUSER command, the current connect group of the user issuing the ADDUSER command is used as the default.

C.1.2 Identify DFSMSrmm to RACF

Before RACF 2.1, the only way to associate a started procedure with a RACF user ID was by coding the RACF started procedures table, ICHRIN03. With RACF 2.1, assigning RACF identities to started procedures has been greatly simplified by the introduction of the RACF STARTED class. You can add or modify security definitions for new and existing started procedures by issuing the RDEFINE and RALTER commands.

C.1.2.1 RACF STARTED class

The STARTED class allows you to assign RACF identities to started procedures dynamically using the RDEFINE and RALTER commands. Resource names in the STARTED class have the format *membername.jobname*. You assign identities such as the RACF user ID and group ID, using fields in the STDATA segment. You can define the started procedure resource, using either a generic profile name or a discrete profile name. A RACF generic profile describes one or more data sets that have a similar name structure. A RACF discrete profile describes a specific data set on a specific volume. In our system we created generic profiles for started procedures. Issue the following RACF commands shown in Figure C-2 to assign RACF identities to the DFSMSrmm started procedure and refresh the in-storage profiles, using the SETROPTS REFRESH command. The SETROPTS GENERIC command is needed only when you define generic profiles.

```
RDEFINE STARTED (DFRMM.*) UACC(NONE) STDATA(USER(DFRMM) GROUP(SYS1))
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
```

Figure C-2 Add DFRMM to RACF started class

Note: If you plan to use the EDGLABEL, EDGXPROC, or the EDGBKUP procedure, you must define the procedures in the RACF STARTED class.

Example C-1 shows a batch job that can be used to add the DFSMSrmm user ID in the RACF started class.

Example: C-1 JCL to add DFSMSrmm user ID in RACF STARTED class

```
RACFRMM  JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************* *
//*  Note
```

```
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ****************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
/* ******************************************************************** */
/* CREATE AN ENTRY IN THE DYNAMIC STARTED PROCEDURE TABLE FOR DFRMM  */
/* ******************************************************************** */
RDEFINE STARTED DFRMM.*           -
        STDATA(USER(DFRMM)        -
                  GROUP(SYS1)     -
                  PRIVELEGED(NO)  -
                  TRACE(NO)       -
                  TRUSTED(NO) )
```

C.1.2.2 ICHRIN03 started procedures table

We do not recommend using the RACF started procedures table (ICHRIN03). The preferred way of adding started procedure users to the RACF database is by using the STARTED class.

C.1.3 Define DFSMSrmm resources to RACF

Define RACF FACILITY class profiles to control access to DFSMSrmm functions protected by the DFSMSrmm resource. The DFSMSrmm resources you protect with RACF profiles in the FACILITY class each have an entity name prefixed with STGADMIN.EDG. Table C-1 lists the DFSMSrmm resources.

If you do not protect DFSMSrmm resources with RACF, an equivalent security product, or you do not have a security product in place, DFSMSrmm provides control for some resources as defined in the *DFSMSrmm Implementation and Customization Guide*, SC26-4932.

Table C-1 DFSMSrmm resources and protected functions

DFSMSrmm resource	Function protected
STGADMIN.EDG.FORCE	Use of the RMM FORCE parameter that can be used in the ADDDATASET, CHANGEDATASET, CHANGEVOLUME and DELETEDATASET subcommand to changing information recorded by DFSMSrmm during O/C/EIV processing
STGADMIN.EDG.HOUSEKEEP	Use of DFSMSrmm inventory management functions.

DFSMSrmm resource	Function protected
STGADMIN.EDG.HOUSEKEEP.REPEXT	Use to authorize the creation of report extract files when no other inventory management function is requested.
STGADMIN.EDG.IGNORE.TAPE.volser	Use of duplicate volume serial numbers and ignored volumes. **Recommendation:** Do not assign an access level to the STGADMIN.EDG.IGNORE.TAPE.RMM.volser resource to any specific user group. Instead, wait until a tape volume that must be ignored by DFSMSrmm is identified, to define a resource granting a user or user group the needed access level. Once the volume is no longer needed, delete the resource.
STGADMIN.EDG.IGNORE.TAPE.RMM.volser	Use of duplicate volume serial numbers and ignoring volumes if the volume is defined to DFSMSrmm. **Recommendation:** Do not assign an access level to the STGADMIN.EDG.IGNORE.TAPE.RMM.volser resource to any specific user group. Instead, wait until a tape volume that must be ignored by DFSMSrmm is identified, to define a resource granting a user or user group the needed access level. Once the volume is no longer needed, delete the resource.
STGADMIN.EDG.IGNORE.TAPE.NORMM.volser	Use of duplicate volume serial numbers and ignoring volumes if the volume is not defined to DFSMSrmm.
STGADMIN.EDG.LABEL.volser	Creation of standard tape labels
STGADMIN.EDG.LISTCONTROL	Use of the RMM LISTCONTROL subcommand to display DFSMSrmm CDS control record information and EDGRMMxx PARMLIB settings
STGADMIN.EDG.MASTER	Access to information in the DFSMSrmm CDS
STGADMIN.EDG.NOLABEL.volser	Creation of tapes without labels
STGADMIN.EDG.OPERATOR	Use of the initialize and erase function
STGADMIN.EDG.OWNER. owner	Access to owned resources. DFSMSrmm checks this entity only if the command issuer is not the owner of the resource or does not have CONTROL access to STDADMIN.EDG.MASTER

Archived

DFSMSrmm resource	Function protected
STGADMIN.EDG.RELEASE	Use of the RMM DELETEVOLUME RELEASE subcommand to process any release actions specified for a volume
STGADMIN.EDG.RESET.SSI	Use of the RESET facility for removing DFSMSrmm from the system
STGADMIN.EDG.VRS	Use of the RMM LISTVRS or SEARCHVRS subcommands to obtain information about VRSs. Use of the RMM ADDVRS and DELETEVRS subcommands to define or remove vital record specifications
STGADMIN.EDG.INERS.WRONGLABEL	Processing for volume mounted with the wrong label using the EDGINERS initialization process

C.1.4 Define RACF groups for DFSMSrmm users

To implement RACF security for DFSMSrmm resources, you should define a set of user groups to handle different kinds of resources at different access levels. Here is our suggestion for function groups:

- **General user**
  They might want to manage volumes they own and request information about resources defined to DFSMSrmm.
- **Storage administrator**
  These persons are responsible for specifying the policies to retain and move data sets and volumes.
- **System programmer**
  The system programmer is responsible for the DFSMSrmm installation and implementation, and provides support for DFSMSrmm in the computing center.
- **Librarian**
  The librarian creates scratch pull lists, moves the tapes between the built-in locations (LOCAL, DISTANT, REMOTE, SHELF), installation defined storage locations, and system managed libraries.
- **Inventory management functions**
  The inventory management function is responsible for the daily and weekly DFSMSrmm inventory management.
- **Operator**
  The operator handles DFSMSrmm system requests including initializing and erasing tapes.

Archive

To implement RACF security for DFSMSrmm resources, we define a RACF group for each of the user groups, connect the user IDs to the appropriate function group, and permit only the RACF groups to the resources in the FACILITY class and to the DFSMSrmm data set profiles.

Table C-2 provides suggestions for authorizing different types of users. Customers should implement security authorization to DFSMSrmm resources according to their working environments.

Table C-2 Access level for each user group

	General User	Storage Administrator	System Pro-grammer	Librarian	Inventory Management Functions	Operator
STGADMIN.EDG.FORCE	-	-	U	U	-	-
STGADMIN.EDG.HOUSEKEEP	-	-	-	-	R	-
STGADMIN.EDG.HOUSEKEEP.REPEXT	-	R	-	R	-	-
STGADMIN.EDG.IGNORE.TAPE. volser	-	-	-	-	-	-
STGADMIN.EDG.IGNORE.TAPE. RMM.volser	-	-	-	-	-	-
STGADMIN.EDG.IGNORE.TAPE.NORMM.volser	-	-	-	-	-	-
STGADMIN.EDG.LABEL. volser	A	-	-	-	-	-
STGADMIN.EDG.LISTCONTROL	C	C	C	C	-	C
STGADMIN.EDG.MASTER	R	C	C	C	-	R
STGADMIN.EDG.NOLABEL. volser	A	-	-	-	-	-
STGADMIN.EDG.OPERATOR	-	-	U	U	-	U
STGADMIN.EDG.OWNER. owner	-	U	-	-	-	-
STGADMIN.EDG.RELEASE	R	-	-	-	-	-
STGADMIN.EDG.RESET.SSI	-	-	-	-	-	A
STGADMIN.EDG.VRS	R	C	C	C	-	-
STGADMIN.EDG.INERS.WRONGLABEL	-	-	-	U	-	-
STGADMIN.ADR.DUMP.CNCURRNT	-	-	-	-	R	-

Where:

A	Alter access
C	Control access
R	Read access
U	Update access
-	Not used

Example C-2 shows a batch job to create the DFSMSrmm RACF groups and connect user IDs to the appropriate group. You can use this JCL as a base and modify it to fit your installation standards.

Example: C-2 JCL to create RACF groups and connect user IDs

```
//RACFGRP  JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************** *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /*   DEFINE RACF ACCESS GROUP FOR ADMINISTRATOR                   */
 /* ************************************************************** */
   ADDGROUP EDGADMIN -
            SUBGROUP(SYS1) OWNER(SYS1)
      CONNECT WOODY    GROUP(EDGADMIN) AUTHORITY(USE)
      CONNECT KUEHN    GROUP(EDGADMIN) AUTHORITY(USE)
      CONNECT GOHR     GROUP(EDGADMIN) AUTHORITY(USE)
      CONNECT ETZ      GROUP(EDGADMIN) AUTHORITY(USE)
      CONNECT BENGT    GROUP(EDGADMIN) AUTHORITY(USE)
      CONNECT NELSON   GROUP(EDGADMIN) AUTHORITY(USE)
 /* ************************************************************** */
 /*   DEFINE RACF ACCESS GROUP FOR SYSTEM PROGRAMMER               */
 /* ************************************************************** */
   ADDGROUP EDGSYSPG -
            SUBGROUP(SYS1) OWNER(SYS1)
      CONNECT SCHLUM   GROUP(EDGSYSPG) AUTHORITY(USE)
      CONNECT SIEGEL   GROUP(EDGSYSPG) AUTHORITY(USE)
 /* ************************************************************** */
 /*   DEFINE RACF ACCESS GROUP FOR LIBRARIAN                       */
 /* ************************************************************** */
   ADDGROUP EDGLIB   -
            SUBGROUP(SYS1) OWNER(SYS1)
      CONNECT JOY      GROUP(EDGLIB) AUTHORITY(USE)
      CONNECT NADINE   GROUP(EDGLIB) AUTHORITY(USE)
      CONNECT ZERBINI  GROUP(EDGLIB) AUTHORITY(USE)
```

Archived

```
     CONNECT STRAUS   GROUP(EDGLIB) AUTHORITY(USE)
     CONNECT SEYFERT  GROUP(EDGLIB) AUTHORITY(USE)
/* ************************************************************** */
/*   DEFINE RACF ACCESS GROUP FOR INVENTORY MANAGEMENT FUNCTION      */
/* ************************************************************** */
  ADDGROUP EDGJOB   -
           SUBGROUP(SYS1) OWNER(SYS1)
     CONNECT WILDEN   GROUP(EDGJOB) AUTHORITY(USE)
     CONNECT HENICKE  GROUP(EDGJOB) AUTHORITY(USE)
     CONNECT KEIL     GROUP(EDGJOB) AUTHORITY(USE)
/* ************************************************************** */
/*   DEFINE RACF ACCESS GROUP FOR INVENTORY MANAGEMENT FUNCTION      */
/* ************************************************************** */
  ADDGROUP EDGOPER -
           SUBGROUP(SYS1) OWNER(SYS1)
     CONNECT TAUBER   GROUP(EDGOPER) AUTHORITY(USE)
     CONNECT ZINK     GROUP(EDGOPER) AUTHORITY(USE)
     CONNECT LOVELACE  GROUP(EDGOPER) AUTHORITY(USE)
```

C.1.4.1 RACF data set profiles

To protect the DFSMSrmm data sets, we define discrete profiles for the following data sets, and then permit the appropriate RACF groups to the following RACF data set profiles:

- DFSMSrmm control data set
- DFSMSrmm journal data set
- DFSMSrmm message data set
- DFSMSrmm housekeeping data sets
- DFSMSrmm report data sets
- DFSMSrmm security report data sets

To protect all other DFSMSrmm data sets, define a generic data set profile using the high level qualifier of your DFSMSrmm data sets.

Note: Refer to the *z/OS Security Server RACF Command Language Reference*, SA22-7687, for enhanced generic naming considerations. The enhanced generic naming option applies only to data sets and allows you to use double asterisk (**) in the the DATASET class. It also changes the meaning of the single asterisk (*) at the end of a profile name.

Example C-3 shows a batch job to define all the DFSMSrmm data set profiles and permit the appropriate RACF groups and DFSMSrmm procedure to the data set profiles.

Archived

Example: C-3 JCL to permit access to DFSMSrmm data set profiles

```
//RACFDSN  JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************** *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /*   DEFINE RACF GROUP                                            */
 /* ************************************************************** */
   ADDGROUP RMM  -
            SUBGROUP(SYS1) OWNER(SYS1)
 /* ************************************************************** */
 /*   DEFINE GLOBAL DATA SET PROFILE                               */
 /* ************************************************************** */
   ADDSD 'RMM.*' UACC(NONE)
      PERMIT 'RMM.*' -
             ID(EDGADMIN) ACC(NONE)
      PERMIT 'RMM.*' -
             ID(EDGLIB) ACC(NONE)
      PERMIT 'RMM.*' -
             ID(EDGJOB) ACC(NONE)
      PERMIT 'RMM.*' -
             ID(EDGOPER) ACC(NONE)
      PERMIT 'RMM.*' -
             ID(EDGSYSPG) ACC(NONE)
 /* ************************************************************** */
 /*   DEFINE CONTROL DATA SET DATA SET PROFILE                     */
 /* ************************************************************** */
   ADDSD 'RMM.PROD.CDS' UACC(NONE) GENERIC
      PERMIT 'RMM.PROD.CDS' GENERIC-
             ID(EDGADMIN) ACC(READ)
      PERMIT 'RMM.PROD.CDS' GENERIC-
             ID(EDGLIB) ACC(READ)
      PERMIT 'RMM.PROD.CDS' GENERIC-
             ID(EDGSYSPG) ACC(ALTER)
 /* ************************************************************** */
 /*   DEFINE JOURNAL DATA SET DATA SET PROFILE                     */
 /* ************************************************************** */
   ADDSD 'RMM.PROD.JRNL' UACC(NONE) GENERIC
      PERMIT 'RMM.PROD.JRNL' GENERIC-
             ID(EDGADMIN) ACC(READ)
      PERMIT 'RMM.PROD.JRNL' GENERIC-
```

Archived

```
            ID(EDGLIB) ACC(READ)
      PERMIT 'RMM.PROD.JRNL' GENERIC-
            ID(EDGSYSPG) ACC(ALTER)
/* ************************************************************** */
/*   DEFINE MESSAGE DATA SET DATA SET PROFILE                     */
/* ************************************************************** */
  ADDSD 'RMM.PROD.MSGS' UACC(NONE) GENERIC
      PERMIT 'RMM.PROD.MSGS' GENERIC-
            ID(EDGADMIN) ACC(UPDATE)
      PERMIT 'RMM.PROD.MSGS' GENERIC-
            ID(EDGLIB) ACC(UPDATE)
      PERMIT 'RMM.PROD.MSGS' GENERIC-
            ID(EDGSYSPG) ACC(ALTER)
/* ************************************************************** */
/*   DEFINE HOUSEKEEP DATA SET DATA SET PROFILE                   */
/* ************************************************************** */
  ADDSD 'RMM.HSKP.*' UACC(NONE)
      PERMIT 'RMM.HSKP.*' -
            ID(EDGADMIN) ACC(ALTER)
      PERMIT 'RMM.HSKP.*' -
            ID(EDGLIB) ACC(ALTER)
      PERMIT 'RMM.HSKP.*' -
            ID(EDGSYSPG) ACC(ALTER)
      PERMIT 'RMM.HSKP.*' -
            ID(EDGOPER) ACC(ALTER)
      PERMIT 'RMM.HSKP.*' -
            ID(EDGJOB) ACC(ALTER)
/* ************************************************************** */
/*   DEFINE REPORT DATA SET DATA SET PROFILE                      */
/* ************************************************************** */
  ADDSD 'RMM.REPORT.*' UACC(NONE)
      PERMIT 'RMM.REPORT.*' -
            ID(EDGADMIN) ACC(ALTER)
      PERMIT 'RMM.REPORT.*' -
            ID(EDGLIB) ACC(ALTER)
      PERMIT 'RMM.REPORT.*' -
            ID(EDGSYSPG) ACC(ALTER)
      PERMIT 'RMM.REPORT.*' -
            ID(EDGOPER) ACC(ALTER)
      PERMIT 'RMM.REPORT.*' -
            ID(EDGJOB) ACC(ALTER)
/* ************************************************************** */
/*   DEFINE SECURITY DATA SET DATA SET PROFILE                    */
/* ************************************************************** */
  ADDSD 'RMM.SECURITY.*' UACC(NONE)
      PERMIT 'RMM.SECURITY.*' -
            ID(EDGSYSPG) ACC(ALTER)
      PERMIT 'RMM.SECURITY.*' -
            ID(EDGJOB) ACC(ALTER)
```

Archived

```
     PERMIT 'RMM.SECURITY.*' -
            ID(AUDITOR) ACC(READ)
/* ************************************************************** */
/*   PERMIT DFRMM TO THE CONTROL DATA SET, JOURNAL DATA SET,      */
/*    EXTRACT DATA SET, ACTIVITY LOG AND REPORT DATA SET          */
/* ************************************************************** */
     PERMIT 'RMM.MASTER.CDS' GENERIC-
            ID(DFRMM) ACC(CONTROL)
     PERMIT 'RMM.MASTER.JOURNAL' GENERIC-
            ID(DFRMM) ACC(UPDATE)
     PERMIT 'RMM.MASTER.MSGS' GENERIC-
            ID(DFRMM) ACC(UPDATE)
     PERMIT 'RMM.HSKP*' -
            ID(DFRMM) ACC(UPDATE)
/*
```

C.1.4.2 RACF resource profiles

The following DFSMSrmm resources, described in Table C-1 on page 650, require RACF profiles in the FACILITY class:

- STGADMIN.EDG.FORCE
- STGADMIN.EDG.HOUSEKEEP
- STGADMIN.EDG.HOUSEKEEP.REPEXT
- STGADMIN.EDG.IGNORE.TAPE.volser
- STGADMIN.EDG.IGNORE.TAPE.RMM.volser
- STGADMIN.EDG.IGNORE.TAPE.NORMM.volser
- STGADMIN.EDG.LABEL.volser
- STGADMIN.EDG.LISTCONTROL
- STGADMIN.EDG.MASTER
- STGADMIN.EDG.NOLABEL.volser
- STGADMIN.EDG.OPERATOR
- STGADMIN.EDG.OWNER.userid
- STGADMIN.EDG.RELEASE
- STGADMIN.EDG.RESET.SSI
- STGADMIN.EDG.VRS
- STGADMIN.EDG.INERS.WRONGLABEL

Example C-4 shows a batch job to define the resource profiles in the RACF FACILITY class and permits the RACF function groups as described in Table C-2 on page 653.

Example: C-4 JCL to Define DFSMSrmm Resources in the FACILOITY CLASS

```
//RACFRDEF JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************ */
```

```
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ************************************************************* */
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
/* ************************************************************** */
/* DEFINE RESOURCE STGADMIN.EDG.FORCE */
/* ************************************************************** */
 RDEFINE FACILITY STGADMIN.EDG.FORCE -
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.FORCE -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.FORCE -
          ID(EDGSYSPG) ACC(READ) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.FORCE -
          ID(EDGLIB) ACC(READ) CLASS(FACILITY)
 /* ************************************************************** */
 /* DEFINE RESOURCE STGADMIN.EDG.HOUSEKEEP                         */
 /* ************************************************************** */
  RDEFINE FACILITY STGADMIN.EDG.HOUSEKEEP -
         UACC(NONE) OWNER(SYS1)
    RALT FACILITY STGADMIN.EDG.HOUSEKEEP -
         GLOBALAUDIT(ALL(UPDATE))
    PERMIT STGADMIN.EDG.HOUSEKEEP -
           ID(EDGSYSPG) ACC(READ) CLASS(FACILITY)
    PERMIT STGADMIN.EDG.HOUSEKEEP -
           ID(EDGLIB ) ACC(READ) CLASS(FACILITY)
    PERMIT STGADMIN.EDG.HOUSEKEEP -
           ID(EDGJOB ) ACC(READ) CLASS(FACILITY)
 /* ************************************************************** */
 /* DEFINE RESOURCE STGADMIN.EDG.HOUSEKEEP.RPTEXT                  */
 /* ************************************************************** */
  RDEFINE FACILITY STGADMIN.EDG.HOUSEKEEP.RPTEXT -
         UACC(NONE) OWNER(SYS1)
    RALT FACILITY STGADMIN.EDG.HOUSEKEEP.RPTEXT -
         GLOBALAUDIT(ALL(UPDATE))
    PERMIT STGADMIN.EDG.HOUSEKEEP.RPTEXT -
           ID(EDGSYSPG) ACC(READ) CLASS(FACILITY)
    PERMIT STGADMIN.EDG.HOUSEKEEP -
           ID(EDGLIB ) ACC(READ) CLASS(FACILITY)
    PERMIT STGADMIN.EDG.HOUSEKEEP.RPTEXT -
           ID(EDGJOB ) ACC(READ) CLASS(FACILITY)
 /* ************************************************************** */
 /* DEFINE RESOURCE STGADMIN.EDG.IGNORE.TAPE.*                     */
 /* ************************************************************** */
  RDEFINE FACILITY STGADMIN.EDG.IGNORE.TAPE.* -
```

Archived

```
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.IGNORE.TAPE.* -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.IGNORE.TAPE.* -
          ID(EDGSYSPG) ACC(READ) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.IGNORE.TAPE.* -
          ID(EDGJOB ) ACC(UPDATE) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.IGNORE.TAPE.NORMM.*             */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.IGNORE.TAPE.NORMM.* -
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.IGNORE.TAPE.NORMM.* -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.IGNORE.TAPE.NORMM.* -
          ID(EDGSYSPG) ACC(READ) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.IGNORE.TAPE.NORMM.* -
          ID(EDGJOB ) ACC(UPDATE) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.IGNORE.TAPE.RMM.*               */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.IGNORE.TAPE.RMM.* -
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.IGNORE.TAPE.RMM.* -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.IGNORE.TAPE.RMM.* -
          ID(EDGSYSPG) ACC(READ) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.IGNORE.TAPE.RMM.* -
          ID(EDGJOB ) ACC(UPDATE) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.LABEL.*                         */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.LABEL.* -
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.LABEL.* -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.LABEL.* -
          ID(EDGLIB) ACC(ALTER) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.LABEL.* -
          ID(EDGSYSPG) ACC(ALTER) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.LABEL.* -
          ID(EDGSYSPG) ACC(ALTER) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.LISTCONTROL                     */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.LISTCONTROL -
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.LISTCONTROL -
        GLOBALAUDIT(ALL(UPDATE))
```

```
   PERMIT STGADMIN.EDG.LISTCONTROL -
          ID(EDGADMIN) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.LISTCONTROL -
          ID(EDGSYSPG) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.LISTCONTROL -
          ID(EDGLIB ) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.LISTCONTROL -
          ID(EDGJOB) ACC(CONTROL) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.MASTER                          */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.MASTER -
        UACC(READ) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.MASTER -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.MASTER -
          ID(EDGADMIN) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.MASTER -
          ID(EDGSYSPG) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.MASTER -
          ID(EDGLIB) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.MASTER -
          ID(EDGJOB) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.MASTER -
          ID(EDGOPER) ACC(CONTROL) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.NOLABEL.*                       */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.NOLABEL.* -
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.NOLABEL.* -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.NOLABEL.* -
          ID(EDGADMIN) ACC(ALTER) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.NOLABEL.* -
          ID(EDGLIB ) ACC(ALTER) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.NOLABEL.* -
          ID(EDGOPER ) ACC(ALTER) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.OPERATOR                        */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.OPERATOR -
        UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.OPERATOR -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.OPERATOR -
          ID(EDGSYSPG) ACC(UPDATE) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.OPERATOR -
          ID(EDGLIB) ACC(UPDATE) CLASS(FACILITY)
```

Archived

```
   PERMIT STGADMIN.EDG.OPERATOR -
          ID(EDGOPER) ACC(UPDATE) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.OWNER.*                         */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.OWNER.* -
         UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.OWNER.* -
        GLOBALAUDIT(ALL(UPDATE))
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.RELEASE                         */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.RELEASE -
         UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.RELEASE -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.RELEASE -
          ID(DFHSM) ACC(READ) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.RESET.SSI                       */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.RESET.SSI -
         UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.RESET.SSI -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.RESET.SSI -
          ID(EDGSYSPG) ACC(ALTER) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.RESET.SSI -
          ID(EDGOPER) ACC(ALTER) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.VRS                             */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.VRS -
         UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.VRS -
        GLOBALAUDIT(ALL(UPDATE))
   PERMIT STGADMIN.EDG.VRS -
          ID(EDGADMIN) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.VRS -
          ID(EDGSYSPG) ACC(CONTROL) CLASS(FACILITY)
   PERMIT STGADMIN.EDG.VRS -
          ID(EDGLIB) ACC(CONTROL) CLASS(FACILITY)
/* ************************************************************ */
/* DEFINE RESOURCE STGADMIN.EDG.INERS.WRONGLABEL                */
/* ************************************************************ */
 RDEFINE FACILITY STGADMIN.EDG.INERS.WRONG.LABEL -
         UACC(NONE) OWNER(SYS1)
   RALT FACILITY STGADMIN.EDG.INERS.WRONG.LABEL -
        GLOBALAUDIT(ALL(UPDATE))
```

Archived

```
PERMIT STGADMIN.EDG.INERS.WRONG.LABEL -
       ID(EDGOPER) ACC(UPDATE) CLASS(FACILITY)
PERMIT STGADMIN.EDG.INERS.WRONG.LABEL -
       ID(EDGSYSPG) ACC(CONTROL) CLASS(FACILITY)
PERMIT STGADMIN.EDG.INERS.WRONG.LABEL -
       ID(EDGLIB) ACC(CONTROL) CLASS(FACILITY)
```

C.1.4.3 Protecting DFSMSdss resources

You can use EDGHSKP or EDGBKUP with DFSMSdss DUMP, or the access method services (AMS) REPRO command to create a backup copy of your DFSMSrmm CDS and journal data set. To use DFSMSdss concurrent copy, you must define or update the profile shown in Figure C-3 in the RACF FACILITY class.

```
STGADMIN.ADR.DUMP.CNCURRNT
```

Figure C-3 Add the concurrent copy resource to RACF

Example C-5 shows a batch job to define the resource in the RACF FACILITY class.

Example: C-5 JCL to define DFSMSdss RACF FACILITY class profile

```
//RACFRDEF JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ******************************************************************* *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ******************************************************************* *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ***************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.CNCURRNT                           */
 /* ***************************************************************** */
  RDEFINE FACILITY STGADMIN.ADR.DUMP.CNCURRNT -
         UACC(NONE) OWNER(SYS1    )
    PERMIT STGADMIN.ADR.DUMP.CNCURRNT-
           ID(EDGADMIN) ACC(READ) CLASS(FACILITY)
```

Archive

C.1.4.4 Authorizing DFSMShsm to DFSMSrmm resources

DFSMSrmm can provide enhanced management functions for the tape volumes that DFSMShsm uses for each of its tape functions. DFSMSrmm treats DFSMShsm like any other tape user, not like a tape manager.

To run DFSMShsm with DFSMSrmm, DFSMShsm procedure names must be defined to RACF. Add the name of the DFSMShsm procedure to the RACF STARTED class or the started procedure table ICHRIN03. Use a DFSMShsm user ID other than the default user ID. Refer to *z/OS DFSMShsm Storage Administrators Guide*, SH35-0421, for information about defining the DFSMShsm and ABARS procedure names.

Before you can use DFSMSrmm with DFSMShsm, you must authorize DFSMShsm to the following resources:

- STGADMIN.EDG.MASTER
- STGADMIN.EDG.RELEASE
- STGADMIN.EDG.OWNER.*hsmid*

If you have multiple DFSMShsm user IDs, for example, in a multi-system environment, and any DFSMShsm ID can return tapes to scratch status or to the DFSMShsm tape pool, you must authorize each DFSMShsm user ID. Define a resource profile STGADMIN.EDG.OWNER.*hsmid* for each DFSMShsm user ID, and give the other DFSMShsm user IDs UPDATE access to it.

When using DFSMSrmm with DFSMShsm, DFSMShsm needs access to the following resources to use scratch tapes:

STGADMIN.EDG.MASTER	READ
STGADMIN.EDG.RELEASE	READ
STGADMIN.EDG.OWNER.*hsmid*	UPDATE

DFSMShsm needs access to the following resources to use a DFSMShsm-managed scratch tape pool:

STGADMIN.EDG.MASTER	UPDATE
STGADMIN.EDG.OWNER.*hsmid*	UPDATE

Example C-6 shows a batch job to add the name of the DFSMShsm procedure in the STARTED class, and permits the DFSMShsm user ID to the required DFSMSrmm RACF FACILITY class profiles.

Example: C-6 JCL to define DFSMShsm to DFSMSrmm resources

```
//RACFHSM  JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************* *
//*  Note
```

```
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ****************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /* Create an entry in the dynamic STARTED procedure table for DFHSM  */
 /* ************************************************************** */
   RDEFINE STARTED DFHSM.* -
          STDATA(USER(DFHSM ) -
                 GROUP(SYS1    ) -
                 PRIVELEGED(NO) -
                 TRACE(NO) -
                 TRUSTED(NO) )
      /* ********************************************************* */
      /*   Permit DFHSM to STGADMIN.EDG.nnnnn    resources         */
      /* ********************************************************* */
      PERMIT STGADMIN.EDG.MASTER -
             ID(DFHSM) ACC(READ) CLASS(FACILITY)
      PERMIT STGADMIN.EDG.RELEASE -
             ID(DFHSM) ACC(READ) CLASS(FACILITY)
      PERMIT STGADMIN.EDG.OWNER.DFHSM*
             ID(DFHSM) ACC(UPDATE) CLASS(FACILITY)
```

C.1.4.5 Authorizing ABARS to DFSMSrmm Resources

To run ABARS with DFSMSrmm, the ABARS procedure names must be defined to RACF. Add the name of the ABARS procedure to the RACF STARTED class or to the started procedure table ICHRIN03. Refer to the *z/OS DFSMShsm Storage Administrators Guide*, SH35-0421, for information about defining the DFSMShsm and ABARS procedure names to RACF.

To use DFSMSrmm with ABARS, you must authorize ABARS to the following resources:

- STGADMIN.EDG.MASTER
- STGADMIN.EDG.RELEASE
- STGADMIN.EDG.OWNER.*abarsid*

If you have multiple ABARS user IDs, for example, in a multi-system environment, and any ABARS ID can return tapes to scratch status, you must authorize each ABARS user ID. Define a RACF resource profile STGADMIN.EDG.OWNER.*abarsid* for each ABARS user ID, and give the other ABARS user IDs UPDATE access to it. This allows one ABARS to release the tapes initially obtained from scratch by another ABARS procedure.

Archived

The ABARS user ID needs access to the following resources to use DFSMSrmm with ABARS:

STGADMIN.EDG.MASTER	READ
STGADMIN.EDG.RELEASE	READ
STGADMIN.EDG.OWNER.*abarsid*	UPDATE

Example C-7 shows a batch job to add the name of the ABARS procedure in the STARTED class, and permit the ABARS user ID to the required DFSMSrmm RACF FACILITY class resources.

Example: C-7 JCL to define ABARS to DFSMSrmm resources

```
//RACFABAR JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************** *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /* Create an entry in the dynamic STARTED procedure table for ABARS  */
 /* ************************************************************** */
   RDEFINE STARTED ABARS.* -
          STDATA(USER(ABARS ) -
                 GROUP(SYS1   ) -
                 PRIVELEGED(NO) -
                 TRACE(NO) -
                 TRUSTED(NO) )
      /* ******************************************************* */
      /*   Permit ABARS to STGADMIN.EDG.nnnnn    resources       */
      /* ******************************************************* */
      PERMIT STGADMIN.EDG.MASTER -
            ID(ABARS) ACC(READ) CLASS(FACILITY)
      PERMIT STGADMIN.EDG.RELEASE -
            ID(ABARS) ACC(READ) CLASS(FACILITY)
      PERMIT STGADMIN.EDG.OWNER.ADSM*
            ID(ABARS) ACC(UPDATE) CLASS(FACILITY)
```

C.1.4.6 Authorizing EDGBKUP to DFSMSrmm resources

You use the OPTION JOURNALFULL(xx) command in PARMLIB member EDGRMMnn to define a percentage full threshold for the journal data set. When DFSMSrmm detects that the journal has reached this threshold, the procedure

Archived

specified on the BACKUPPROC operand is started automatically. In this redbook we use the name EDGBKUP for the backup procedure name.

The EDGBKUP procedure requires READ access to the resources shown in Figure C-4, and UPDATE and ALTER access to the data set profiles shown in Figure C-5 to use the backup housekeeping function.

```
STGADMIN.EDG.HOUSEKEEP
STGADMIN.ADR.DUMP.CNCURRNT
```

Figure C-4 HOUSEKEEP resource for EDGBKUP

```
RMM.HSKP.*          ALTER
RMM.PROD.MSG        UPDATE
```

Figure C-5 RMM data set profiles for EDGBKUP

Example C-8 shows a batch job to add the EDGBKUP procedure in the STARTED class, and permit it to the required RACF FACILITY class resources.

Example: C-8 JCL to define EDGBKUP to DFSMSrmm resources

```
//RACFBKUP JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ******************************************************************* *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ******************************************************************* *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /* Create an entry in the STARTED procedure table for EDGBKUP     */
 /* ************************************************************** */
   RDEFINE STARTED EDGBKUP.* -
           STDATA(USER(DFRMM ) -
                  GROUP(SYS1    ) -
                  PRIVELEGED(NO) -
                  TRACE(NO) -
                  TRUSTED(NO) )
      /* ********************************************************** */
      /*   Permit EDGHSKP to STGADMIN.EDG.nnnnn    resources        */
      /* ********************************************************** */
      PERMIT STGADMIN.EDG.HOUSEKEEP -
```

Archived

```
       ID(EDGBKUP) ACC(READ) CLASS(FACILITY)
PERMIT 'RMM.PROD.MSGS' GENERIC -
       ID(EDGBKUP) ACC(UPDATE)
PERMIT 'RMM.HSKP.*' -
       ID(EDGBKUP) ACC(ALTER)
PERMIT STGADMIN.ADR.DUMP.CNCURRNT -
       ID(EDGBKUP) ACC(READ) CLASS(FACILITY)
```

C.1.4.7 Authorizing EDGXPROC to DFSMSrmm resources

You use the OPTION SCRATCHPROC command in PARMLIB member EDGRMMnn to specify the name of the procedure that you want to be started automatically to replenish scratch volumes in an automated tape library dataserver (IBM 3494 or IBM 3495). In this redbook we use the default procedure name of EDGXPROC by specifying SCRATCHPROC(EDGXPROC) in EDGRMMnn. A sample EDGXPROC procedure is shipped with DFSMSrmm.

When a manual or automatic tape library dataserver detects a low-on-scratch condition, where more scratch volumes are needed, OAM issues a write-to-operator message. DFSMSrmm intercepts the message, and starts the procedure you specify with SCRATCHPROC value in PARMLIB.

The EDGXPROC procedure requires READ access to the resources shown in Figure C-6, and UPDATE access to the data set profile shown in Figure C-7 to use the expiration processing housekeeping function.

```
STGADMIN.EDG.HOUSEKEEP
```

Figure C-6 HOUSEKEEP resource for EDGXPROC

```
RMM.PROD.MSG
```

Figure C-7 RMM data set profiles for EDGXPROC

Example C-9 shows a batch job to add the EDGXPROC procedure in the STARTED class and permit it to the required RACF FACILITY class resources.

Example: C-9 JCL to define EDGXPROC to DFSMSrmm resources

```
//RACFXPRC JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************* *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
```

```
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /* Create an entry in the dynamic STARTED procedure table for ABARS  */
 /* ************************************************************** */
   RDEFINE STARTED EDGXPROC.* -
           STDATA(USER(DFRMM ) -
                  GROUP(SYS1     ) -
                  PRIVELEGED(NO) -
                  TRACE(NO) -
                  TRUSTED(NO) )
      /* ********************************************************** */
      /*   Permit EDGXPROC to STGADMIN.EDG.nnnnn    resources       */
      /* ********************************************************** */
      PERMIT STGADMIN.EDG.HOUSEKEEP -
             ID(EDGXPROC) ACC(READ) CLASS(FACILITY)
      PERMIT 'RMM.PROD.MSGS' GENERIC-
             ID(EDGXPROC) ACC(ALTER)
```

C.1.4.8 Authorizing EDGLABEL to DFSMSrmm resources

You can run the EDGINERS utility as an operator-started procedure, so that the operator or librarian can make requests for tape labeling and erasing. The name of the procedure can be any valid alphanumeric procedure name from 1 to 8 characters. In this book we use EDGLABEL as the procedure name.

To automatically initialize or erase tapes, all volumes must be defined in the DFSMSrmm CDS, and the requested action must be set. In the automatic processing mode, EDGINERS requests that DFSMSrmm scan its CDS for the first 10 volumes waiting to be initialized or erased. If no volumes are found, the procedure ends without an error.

Note: DFSMSrmm erases volumes using the hardware security erase feature when supported.

The EDGLABEL procedure requires UPDATE access to the resource shown in Figure C-8 to use the expiration processing housekeeping function.

```
STGADMIN.EDG.OPERATOR
```

Figure C-8 OPERATOR resource for EDGLABEL

Example C-10 shows a batch job to add the EDGLABEL procedure in the STARTED class and permit it to the required RACF FACILITY class resources.

Example: C-10 JCL to define EDGLABEL to DFSMSrmm resources

```
//RACFLBL  JOB ,RMM,NOTIFY=SCHLUM,
//         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
//* ************************************************************** *
//*  Note
//*  ====
//*    To execute this job you need the RACF SPECIAL attribute
//* ************************************************************** *
//STEP01   EXEC PGM=IKJEFT01
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 /* ************************************************************** */
 /* Create an entry in the dynamic STARTED procedure table for LABEL  */
 /* ************************************************************** */
   RDEFINE STARTED EDGLABEL.* -
           STDATA(USER(DFRMM ) -
                  GROUP(SYS1     ) -
                  PRIVELEGED(NO) -
                  TRACE(NO) -
                  TRUSTED(NO) )
      /* ********************************************************** */
      /*   Permit EDGLABEL to STGADMIN.EDG.nnnnn    resources        */
      /* ********************************************************** */
      PERMIT STGADMIN.EDG.OPERATOR  -
           ID(EDGLABEL) ACC(READ) CLASS(FACILITY)
```

C.2 CA-Top Secret implementation

The following sections include the necessary definitions for CA-Top Secret product.

C.2.1 Define user groups

Example C-11 shows the definitions of the different user groups for tape management.

Example: C-11 CA-Top Secret user groups definitions

```
/* ************************************************************** */
/*   DEFINE CA-Top Secret ACCESS GROUP FOR ADMINISTRATOR          */
/* ************************************************************** */
```

```
TSS CRE(EDGADMIN) DEPT(XXXX) NAME('EDG ADMIN') TYPE(PROFILE)
TSS ADD(WOODY ) PROFILE(EDGADMIN)
TSS ADD(KUEHN ) PROFILE(EDGADMIN)
TSS ADD(GOGR  ) PROFILE(EDGADMIN)
TSS ADD(ETZ   ) PROFILE(EDGADMIN)
TSS ADD(BENGT ) PROFILE(EDGADMIN)
TSS ADD(NELSON) PROFILE(EDGADMIN)
 /* ************************************************************** */
 /*   DEFINE CA-Top Secret ACCESS GROUP FOR SYSTEM PROGRAMMER      */
 /* ************************************************************** */
TSS CRE(EDGSYSPG) DEPT(XXXX) NAME('EDG SYSPG') TYPE(PROFILE)
TSS ADD(SCHLUM  ) PROFILE(EDGSYSPG)
TSS ADD(SIEGEL  ) PROFILE(EDGSYSPG)
 /* ************************************************************** */
 /*   DEFINE CA-Top Secret ACCESS GROUP FOR LIBRARIAN              */
 /* ************************************************************** */
TSS CRE(EDGLIB) DEPT(XXXX) NAME('EDG LIBRARIAN') TYPE(PROFILE)
TSS ADD(JOY     ) PROFILE(EDGLIB)
TSS ADD(NADINE  ) PROFILE(EDGLIB)
TSS ADD(ZERBINI ) PROFILE(EDGLIB)
TSS ADD(STRAUS  ) PROFILE(EDGLIB)
TSS ADD(SEYFERT ) PROFILE(EDGLIB)
 /* ************************************************************** */
 /*   DEFINE CA-Top Secret GROUP FOR INVENTORY MANAGEMENT FUNCTION */
 /* ************************************************************** */
TSS CRE(EDGJOB) DEPT(XXXX) NAME('EDG JOB') TYPE(PROFILE)
TSS AD(HENICKE ) PROFILE(EDGJOB)
TSS AD(KEIL    ) PROFILE(EDGJOB)
TSS AD(BRIAN   ) PROFILE(EDGJOB)
 /* ************************************************************** */
 /*   DEFINE CA-Top Secret ACCESS GROUP FOR OPERATOR               */
 /* ************************************************************** */
TSS CRE(EDGOPER) DEPT(XXXX) NAME('EDG OPERATOR') TYPE(PROFILE)
TSS ADD(LOVELACE) PROFILE(EDGOPER)
TSS ADD(TAUBER  ) PROFILE(EDGOPER)
TSS ADD(ZINK    ) PROFILE(EDGOPER)
```

Protect DFSMSrmm data sets
Example shows the profile definitions for the different DFSMSrmm data sets.
CA-Top Secret data sets profile definition

```
 /* ************************************************************** */
 /*   DEFINE GLOBAL DATA SET PROFILE                               */
 /* ************************************************************** */
TSS ADD(XXXX) DSN(RMM.)
TSS PE(EDGADMIN) DSN(RMM.*) ACCESS(NONE)
TSS PE(EDGLIB  ) DSN(RMM.*) ACCESS(NONE)
TSS PE(EDGJOB  ) DSN(RMM.*) ACCESS(NONE)
TSS PE(EDGOPER ) DSN(RMM.*) ACCESS(NONE)
TSS PE(EDGSYSPG) DSN(RMM.*) ACCESS(ALL)
 /* ************************************************************** */
 /*   DEFINE CONTROL DATA SET DATA SET PROFILE                     */
 /* ************************************************************** */
TSS PE(EDGADMIN) DSN(RMM.EGZB.CDS) ACCESS(UPDATE)
```

Archived

```
TSS PE(EDGLIB  ) DSN(RMM.EGZB.CDS) ACCESS(UPDATE)
TSS PE(EDGJOB  ) DSN(RMM.EGZB.CDS) ACCESS(UPDATE)
TSS PE(EDGOPER ) DSN(RMM.EGZB.CDS) ACCESS(UPDATE)
TSS PE(EDGSYSPG) DSN(RMM.EGZB.CDS) ACCESS(ALL)
 /* ************************************************************** */
 /*   DEFINE JOURNAL DATA SET DATA SET PROFILE                     */
 /* ************************************************************** */
TSS PE(EDGADMIN) DSN(RMM.EGZB.JRNL) ACCESS(UPDATE)
TSS PE(EDGLIB  ) DSN(RMM.EGZB.JRNL) ACCESS(UPDATE)
TSS PE(EDGJOB  ) DSN(RMM.EGZB.JRNL) ACCESS(UPDATE)
TSS PE(EDGOPER ) DSN(RMM.EGZB.JRNL) ACCESS(UPDATE)
TSS PE(EDGSYSPG) DSN(RMM.EGZB.JRNL) ACCESS(ALL)
 /* ************************************************************** */
 /*   DEFINE MESSAGE DATA SET DATA SET PROFILEDEFINE MESSAGE DATA S   */
 /* ************************************************************** */
TSS PE(EDGADMIN) DSN(RMM.HSKP.MESSAGE.DSET) ACCESS(UPDATE)
TSS PE(EDGLIB  ) DSN(RMM.HSKP.MESSAGE.DSET) ACCESS(UPDATE)
TSS PE(EDGJOB  ) DSN(RMM.HSKP.MESSAGE.DSET) ACCESS(UPDATE)
TSS PE(EDGOPER ) DSN(RMM.HSKP.MESSAGE.DSET) ACCESS(UPDATE)
TSS PE(EDGSYSPG) DSN(RMM.HSKP.MESSAGE.DSET) ACCESS(ALL)
 /* ************************************************************** */
 /*   DEFINE HOUSEKEEP DATA SET DATA SET PROFILE                   */
 /* ************************************************************** */
TSS PE(EDGSYSPG) DSN(RMM.HSKP.*) ACCESS(ALL)
TSS PE(EDGLIB  ) DSN(RMM.HSKP.*) ACCESS(ALL)
TSS PE(EDGJOB  ) DSN(RMM.HSKP.*) ACCESS(ALL)
TSS PE(EDGOPER ) DSN(RMM.HSKP.*) ACCESS(ALL)
 /* ************************************************************** */
 /*   DEFINE REPORT DATA SET DATA SET PROFILE                      */
 /* ************************************************************** */
TSS PE(EDGSYSPG) DSN(RMM.REPORT.*) ACCESS(ALL)
TSS PE(EDGSYSPG) DSN(RMM.REPORT.*) ACCESS(ALL)
TSS PE(EDGSYSPG) DSN(RMM.REPORT.*) ACCESS(ALL)
TSS PE(EDGSYSPG) DSN(RMM.REPORT.*) ACCESS(ALL)
```

Define DFSMSrmm resources
Example shows the definitions for the DFSMSrmm resources.
CA-Top Secret DFSMSrmm resource definition

```
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.HOUSEKEEP                       */
 /* ************************************************************** */
TSS ADD(XXXX) IBMFAC(STGADMIN)     IF STGADMIN not already defined
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.IGNORE.HOUSEKEEP) ACCESS(READ) ACTION(AUDIT)
TSS PE(EDGJOB  ) IBMFAC(STGADMIN.EDG.IGNORE.HOUSEKEEP) ACCESS(READ) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.IGNORE.TAPE.*                   */
 /* ************************************************************** */
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.IGNORE.TAPE.*) ACCESS(UPDATE) ACTION(AUDIT)
TSS PE(EDGJOB  ) IBMFAC(STGADMIN.EDG.IGNORE.TAPE.*) ACCESS(UPDATE) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.IGNORE.TAPE.RMM.*                  */
 /* ************************************************************** */
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.IGNORE.TAPE.RMM.*) ACCESS(UPDATE) ACTION(AUDIT)
```

```
TSS PE(EDGJOB  ) IBMFAC(STGADMIN.EDG.IGNORE.TAPE.RMM*) ACCESS(UPDATE) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.IGNORE.TAPE.NORMM.*                 */
 /* ************************************************************** */
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.IGNORE.TAPE.NORMM.*) ACCESS(UPDATE) ACTION(AUDIT)
TSS PE(EDGJOB ) IBMFAC(STGADMIN.EDG.IGNORE.TAPE.NORMM.*) ACCESS(UPDATE) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.LABEL.*                         */
 /* ************************************************************** */
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.LABEL.*) ACCESS(ALL) ACTION(AUDIT)
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.LABEL.*) ACCESS(ALL) ACTION(AUDIT)
TSS PE(EDGJOB  ) IBMFAC(STGADMIN.EDG.LABEL.*) ACCESS(ALL) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.LISTCONTROL                     */
 /* ************************************************************** */
TSS PE(EDGADMIN) IBMFAC(STGADMIN.EDG.LISTCONTROL) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.LISTCONTROL) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.LISTCONTROL) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGJOB  ) IBMFAC(STGADMIN.EDG.LISTCONTROL) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGOPER ) IBMFAC(STGADMIN.EDG.LISTCONTROL) ACCESS(CONTROL) ACTION(AUDIT)
  /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.MASTER                          */
 /* ************************************************************** */
TSS PE(EDGADMIN) IBMFAC(STGADMIN.EDG.MASTER) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.MASTER) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.MASTER) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGJOB  ) IBMFAC(STGADMIN.EDG.MASTER) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGOPER ) IBMFAC(STGADMIN.EDG.MASTER) ACCESS(CONTROL) ACTION(AUDIT)
/* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.NOLABEL.*                       */
 /* ************************************************************** */
TSS PE(EDGADMIN) IBMFAC(STGADMIN.EDG.NOLABEL.*) ACCESS(ALL) ACTION(AUDIT)
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.NOLABEL.*) ACCESS(ALL) ACTION(AUDIT)
TSS PE(EDGOPER ) IBMFAC(STGADMIN.EDG.NOLABEL.*) ACCESS(ALL) ACTION(AUDIT)
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.OPERATOR                        */
/* ************************************************************** */
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.OPERATOR) ACCESS(UPDATE) ACTION(AUDIT)
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.OPERATOR) ACCESS(UPDATE) ACTION(AUDIT)
TSS PE(EDGOPER ) IBMFAC(STGADMIN.EDG.OPERATOR) ACCESS(UPDATE) ACTION(AUDIT)
/* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.OWNER.*                         */
 /* ************************************************************** */
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.OWNER.*) ACCESS(UPDATE) ACTION(AUDIT)
TSS PE(EDGADMIN) IBMFAC(STGADMIN.EDG.OWNER.*) ACCESS(UPDATE) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.RELEASE                         */
 /* ************************************************************** */
TSS PE(DFHSM   ) IBMFAC(STGADMIN.EDG.RELEASE) ACCESS(READ) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.RESET.SSI                       */
 /* ************************************************************** */
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.RESET.SSI) ACCESS(ALTER) ACTION(AUDIT)
TSS PE(EDGOPER ) IBMFAC(STGADMIN.EDG.RESET.SSI) ACCESS(ALTER) ACTION(AUDIT)
```

```
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.VRS                             */
 /* ************************************************************** */
TSS PE(EDGADMIN) IBMFAC(STGADMIN.EDG.VRS) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.VRS) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.VRS) ACCESS(CONTROL) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.INERS.WRONGLABEL                */
 /* ************************************************************** */
TSS PE(EDGOPER ) IBMFAC(STGADMIN.EDG.INERS.WRONGLABEL) ACCESS(UPDATE) ACTION(AUDIT)
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.EDG.INERS.WRONGLABEL) ACCESS(CONTROL) ACTION(AUDIT)
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.EDG.INERS.WRONGLABEL) ACCESS(CONTROL) ACTION(AUDIT)
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.ADR.DUMP.CNCURRNT                   */
 /* ************************************************************** */
TSS PE(EDGSYSPG) IBMFAC(STGADMIN.ADR.DUMP.CNCURRNT) ACCESS(READ) ACTION(AUDIT)
TSS PE(EDGLIB  ) IBMFAC(STGADMIN.ADR.DUMP.CNCURRNT) ACCESS(READ) ACTION(AUDIT)
TSS PE(EDGJOB  ) IBMFAC(STGADMIN.ADR.DUMP.CNCURRNT) ACCESS(READ) ACTION(AUDIT)
```

C.3 ACF2 implementation

The following sections include the necessary definitions for the ACF2 product.

C.3.1 Define data set rules

The definition rules for DFSMSrmm data sets are shown in Example C-12.

Example: C-12 ACF2 data set rules definition

```
    ADDGROUP EDGADMIN  =  UIDSTRING1
    ADDGROUP EDGSYSPG  =  UIDSTRING2
    ADDGROUP EDGLIB    =  UIDSTRING3
    ADDGROUP EDGJOB    =  UIDSTRING4
    ADDGROUP EDGOPER   =  UIDSTRING5
/* ************************************************************** */
/*   DEFINE DATA SET RULE FOR HLQ 'RMM'                           */
/* ************************************************************** */
  $KEY(RMM)
   EGZB.CDS UID(UIDSTRING1) E(A) R(A) W(A)
   EGZB.CDS UID(UIDSTRING3) E(A) R(A) W(A)
   EGZB.CDS UID(UIDSTRING4) E(A) R(A) W(A)
   EGZB.CDS UID(UIDSTRING5) E(A) R(A) W(A)
   EGZB.CDS UID(UIDSTRING2) E(A) R(A) W(A) A(A)
   EGZB.JRNL UID(UIDSTRING1) E(A) R(A) W(A)
   EGZB.JRNL UID(UIDSTRING3) E(A) R(A) W(A)
   EGZB.JRNL UID(UIDSTRING4) E(A) R(A) W(A)
   EGZB.JRNL UID(UIDSTRING5) E(A) R(A) W(A)
   EGZB.JRNL UID(UIDSTRING2) E(A) R(A) W(A) A(A)
```

Archive

```
HSKP.MESSAGE.DSET UID(UIDSTRING1) E(A) R(A) W(A)
HSKP.MESSAGE.DSET UID(UIDSTRING3) E(A) R(A) W(A)
HSKP.MESSAGE.DSET UID(UIDSTRING4) E(A) R(A) W(A)
HSKP.MESSAGE.DSET UID(UIDSTRING5) E(A) R(A) W(A)
HSKP.MESSAGE.DSET UID(UIDSTRING2) E(A) R(A) W(A) A(A)
HSKP.* UID(UIDSTRING3) E(A) R(A) W(A) A(A)
HSKP.* UID(UIDSTRING4) E(A) R(A) W(A) A(A)
HSKP.* UID(UIDSTRING5) E(A) R(A) W(A) A(A)
HSKP.* UID(UIDSTRING2) E(A) R(A) W(A) A(A)
REPORT.* UID(UIDSTRING3) E(A) R(A) W(A) A(A)
REPORT.* UID(UIDSTRING4) E(A) R(A) W(A) A(A)
REPORT.* UID(UIDSTRING5) E(A) R(A) W(A) A(A)
REPORT.* UID(UIDSTRING2) E(A) R(A) W(A) A(A)
* UID(*) UID(UIDSTRING2) E(A) R(A) W(A) A(A)
* UID(*)
```

C.3.2 Define DFSMSrmm resources

The definitions for the DFSMSrmm resources are shown in Example C-13.

Example: C-13 ACF2 DFSMSrmm resources definition

```
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.HOUSEKEEP                       */
 /* ************************************************************** */
$KEY(STGADMIN.EDG.HOUSEKEEP) TYPE(FAC)
UID(UIDSTRING2) SERVICE(READ) ALLOW
UID(UIDSTRING4) SERVICE(READ) ALLOW
UID(UIDSTRING5) SERVICE(READ) ALLOW
/* ************************************************************** */
/*   DEFINE ACF2 RESOURCE STGADMIN.EDG.IGNORE.TAPE.VOLSER.*
/ * ************************************************************** */
$KEY(STGADMIN.EDG.IGNORE.TAPE.VOLSER.*) TYPE(FAC)
UID(UIDSTRING2) SERVICE(READ) ALLOW
UID(UIDSTRING4) SERVICE(READ,UPDATE) LOG
/* ************************************************************** */
/*   DEFINE ACF2 RESOURCE STGADMIN.EDG.IGNORE.TAPE.RMM.VOLSER.*
/ * ************************************************************** */
$KEY(STGADMIN.EDG.IGNORE.TAPE.RMM.VOLSER.*) TYPE(FAC)
UID(UIDSTRING2) SERVICE(READ) ALLOW
UID(UIDSTRING4) SERVICE(READ,UPDATE) LOG
/* ************************************************************** */
/*   DEFINE ACF2 RESOURCE STGADMIN.EDG.IGNORE.TAPE.NORMM.VOLSER.*
/ * ************************************************************** */
$KEY(STGADMIN.EDG.IGNORE.TAPE.NORMM.VOLSER.*) TYPE(FAC)
UID(UIDSTRING2) SERVICE(READ) ALLOW
UID(UIDSTRING4) SERVICE(READ,UPDATE) LOG
/* ************************************************************** */
```

Archived

```
/*   DEFINE ACF2 RESOURCE STGADMIN.EDG.LABEL.*
/* ************************************************************** */
$KEY(STGADMIN.EDG.LABEL.*) TYPE(FAC)
UID(UIDSTRING1) LOG
UID(UIDSTRING2) LOG
UID(UIDSTRING5) LOG
/* ************************************************************** */
/*   DEFINE ACF2 RESOURCE STGADMIN.EDG.LIST.CONTROL
/* ************************************************************** */
$KEY(STGADMIN.EDG.LISTCONTROL) TYPE(FAC)
UID(UIDSTRING1) SERVICE(UPDATE) LOG
UID(UIDSTRING2) SERVICE(UPDATE) LOG
UID(UIDSTRING3) SERVICE(UPDATE) LOG
UID(UIDSTRING4) SERVICE(UPDATE) LOG
UID(UIDSTRING5) SERVICE(UPDATE) LOG
/* ************************************************************** */
/*   DEFINE ACF2 RESOURCE STGADMIN.EDG.MASTER
/* ************************************************************** */
$KEY(STGADMIN.EDG.MASTER) TYPE(FAC)
UID(UIDSTRING1) SERVICE(UPDATE) LOG
UID(UIDSTRING2) SERVICE(UPDATE) LOG
UID(UIDSTRING3) SERVICE(UPDATE) LOG
UID(UIDSTRING4) SERVICE(UPDATE) LOG
UID(UIDSTRING5) SERVICE(UPDATE) LOG
 /* ************************************************************** */
 /*   DEFINE RESOURCE STGADMIN.EDG.NOLABEL.*                       */
 /* ************************************************************** */
$KEY(STGADMIN.EDG.NOLABEL.*) TYPE(FAC)
UID(UIDSTRING1) LOG
UID(UIDSTRING3) LOG
UID(UIDSTRING5) LOG
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.MASTER                          */
/* ************************************************************** */
$KEY(STGADMIN.EDG.MASTER) TYPE(FAC)
UID(UIDSTRING2) SERVICE(READ,UPDATE) LOG
UID(UIDSTRING3) SERVICE(READ,UPDATE) LOG
UID(UIDSTRING5) SERVIVE(READ,UPDATE) LOG
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.OWNER.*                         */
/* ************************************************************** */
$KEY(STGADMIN.EDG.OWNER) TYPE(FAC)
UID(UIDSTRING1) SERVICE(READ,UPDATE) LOG
UID(UIDSTRING3) SERVICE(READ,UPDATE) LOG
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.RELEASE                         */
/* ************************************************************** */
$KEY(STGADMIN.EDG.RELEASE) TYPE(FAC)
UID(DFHSM) SERVICE(READ) ALLOW
```

Arrived

```
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.RESET.SSI                       */
/* ************************************************************** */
$KEY(STGADMIN.EDG.RESET.SSI) TYPE(FAC)
UID(UIDSTRING2) LOG
UID(UIDSTRING5) LOG
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.VRS                             */
/* ************************************************************** */
$KEY(STGADMIN.EDG.VRS) TYPE(FAC)
UID(UIDSTRING1) SERVICE(READ,UPDATE) LOG
UID(UIDSTRING2) SERVICE(READ,UPDATE) LOG
UID(UIDSTRING3) SERVICE(READ,UPDATE) LOG
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.INERS.WRONGLABEL                */
/* ************************************************************** */
$KEY(STGADMIN.EDG.INERS.WRONGLABEL) TYPE(FAC)
UID(UIDSTRING1) LOG
UID(UIDSTRING2) LOG
UID(UIDSTRING5) SERVICE(READ,UPDATE) LOG
/* ************************************************************** */
/*   DEFINE ACF2 RESOURCE STGADMIN.ADR.DUMP.CNCURRNT              */
/* ************************************************************** */
$KEY(STGADMIN.EDG.INERS.WRONGLABEL) TYPE(FAC)
UID(UIDSTRING1) SERVICE(READ) ALLOW
UID(UIDSTRING2) SERVICE(READ) ALLOW
UID(UIDSTRING4) SERVICE(READ) ALLOW
```

Archived

Archived

D

Additional material

This redbook refers to additional material that can be downloaded from the Internet as described below.

Locating the Web material

The Web material associated with this redbook is available in softcopy on the Internet from the IBM Redbooks Web server. Point your Web browser to:

ftp://www.redbooks.ibm.com/redbooks/SG246275

Alternatively, you can go to the IBM Redbooks Web site at:

ibm.com/redbooks

Select the **Additional materials** and open the directory that corresponds with the redbook form number, SG246275.

Using the Web material

The additional Web material that accompanies this redbook includes the following file:

File name	*Description*
samples.zip	Zipped sample exits

© Copyright IBM Corp. 2004. All rights reserved.

System requirements for downloading the Web material

The following system configuration is recommended:

Hard disk space: 1 MB minimum
Operating System: Any
Processor: Any
Memory: Any

How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder. Read the file README.TXT for more information.

Archived

Glossary

A

abend. Abnormal end of task.

AL. American National Standards Label.

AMODE. Addressing mode.

AMS. Access method services.

ANDVRS. An RMM ADDVRS TSO subcommand operand. See Using AND.

ANSI. American National Standards Institute.

APAR. Authorized program analysis report.

APF. Authorized program facility.

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.

ATL. Automated tape library.

AUL. ANSI and user header or trailed label.

automated tape library. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. See also tape library. Contrast with manual tape library.

automatic cartridge loader. An optional feature of the 3480 Magnetic Tape Subsystem that allows reloading of multiple tape cartridges. This feature is standard in the 3490 Magnetic Tape Subsystem.

automatic recording. In DFSMSrmm, the process of recording information about a volume and the data sets on the volume in the DFSMSrmm control data set at open or close time.

availability. For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

B

backup. The process of creating a copy of a data set or object to be used in case of accidental loss.

basic catalog structure. The name of the catalog structure in the integrated catalog facility environment. See also integrated catalog facility catalog.

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 DFSMSrmm defined storage locations: LOCAL, DISTANT, and REMOTE.

C

cache fast write. A storage control capability in which the data is written directly to cache without using non-volatile storage. Cache fast write is useful for temporary data that is readily recreated., such as the sort work files created by DFSORT. Contrast with DASD fast write.

© Copyright IBM Corp. 2004. All rights reserved.

cartridge eject. For an IBM 3494 or IBM 3495 Tape Library Data set, the act of physically removing a tape cartridge usually under robot control, by placing it in an output station. The software logically removes the cartridge by deleting or updating the tape volume record in the tape configuration database. For a manual tape library dataserver, the act of logically removing a tape cartridge from the manual tape library dataserver by deleting or updating the tape volume record in the tape configuration database.

cartridge entry. For either an IBM 3494 Tape Library Dataserver, IBM 3495 Tape Library Dataserver, or a IBM Model M10 3495 Tape Library Dataserver, the process of logically adding a tape cartridge to the library by creating or updating the tape volume record in the tape configuration database. The cartridge entry process includes the assignment of the cartridge to scratch or private category in the library.erm2 definition.

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.

class. A collection of RACF-defined entities (users, groups, and resources) with similar characteristics. Classes are defined in the class descriptor table (CDT), except for the USER, GROUP, and DATASET classes.

class authority (CLAUTH). An attribute enabling a user to define RACF profiles in a class defined in the class descriptor table. A user can have class authority to zero or more classes.

CLAUTH attribute. See class authority.

conditional access list. The portion of a resource profile that specifies the users and groups that may access the resource at a specified level when a specified condition is true. For example, with program access to data sets, the condition is that the user must be executing the program specified in the access list. Contrast with standard access list.

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.

control data set. A VSAM key-sequenced data set that contains the complete inventory of your removable media library, as well as the movement and retention policies you define. In the control data set DFSMSrmm records all changes made to the inventory, such as adding or deleting volumes.

control data set ID. A one-to-eight character identifier for the DFSMSrmm control data set used to ensure that, in a multi-system, multi-complex environment, the correct management functions are performed.

convenience input. The process of adding a small number of tape cartridges to the IBM 3494 Tape Library Dataserver and IBM 3495 Tape Library Dataserver without interrupting operations, by inserting the cartridges directly into cells in a convenience input station.

convenience input/output station. A transfer station with combined tape cartridge input and output functions in the IBM 3494 Tape Library Dataservers only.

convenience input station. A transfer station, used by the operator to add tape cartridges to the IBM 3494 Tape Library Dataserver or an IBM 3495 Tape Library Dataserver, which is accessible from outside the enclosure area.

convenience output. The process of removing a small number of tape cartridges from the IBM 3494 Tape Library Dataserver or an IBM 3495 Tape Library Dataserver without interrupting operations, by removing the cartridges directly from cells in a convenience input station.

convenience output station. A transfer station, used by the operator to remove tape cartridges from the automated tape library dataserver, which is accessible from outside the enclosure area.

conversion. In DFSMSrmm, the process of moving your removable media library inventory from another media management system to DFSMSrmm. DFSMSrmm manages the inventory and policies once you have converted it.

create date. Create date for a dataset is the date that the dataset is written to tape. Create date can also be the date a data set was read if it was created before DFSMSrmm is in use. Create date is updated each time a data set is replaced and not extended. Create date for volumes and other resources defined to DFSMSrmm is the date the resource is defined to DFSMSrmm or the date specified on the command as the create date.

CST. Cartridge System Tape.

D

data set profile. A profile that provides RACF protection for one or more data sets. The information in the profile can include the data set profile name, profile owner, universal access authority, access list, and other data. See discrete profile and generic profile.

DASD. Direct access storage device.

DASF fast write. An extended function of some models of the IBM 3990 Storage Control in which data is written concurrently to cache and nonvolatile storage and automatically scheduled for destaging to DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. Use of DASD fast write for system-managed data sets is controlled by storage class attributes to improve performance. See also dynamic cache management. Contrast with cache fast write.

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also volume, primary storage, migration level 1, migration level 2.

data column. A vertical arrangement of identical data items, used on list panels to display an attribute, characteristic, or value of one or more objects.

data entry panel. A panel in which the user communicates with the system by filling in one or more fields.

DCB. Data control block.

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.

device. This term is used interchangeably with unit. You mount a tape on a unit or device, such as a 3490.

DFSMS environment. An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the DFSMS environment for MVS, the function is provided by DFSORT, RACF, and the combination of DFSMS/MVS and MVS.

DFSMS/MVS. An IBM System/390® licensed program that provides storage, data, and device management functions. When combined with MVS/ESA™ SP™ Version 5 it composes the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm.

DFSMSdfp. A DFSMS/MVS functional component or base element of OS/390, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS/MVS functional component or base element of OS/390, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS/MVS functional component or base element of OS/390, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMShsm-managed volume. (1) A primary storage volume, which is defined to DFSMShsm but which does not belong to a storage group. (2) A volume in a storage group, which is using DFSMShsm automatic dump, migration, or backup services. Contrast with system-managed volume and DFSMSrmm-managed volume.

DFSMShsm-owned volume. A storage volume on which DFSMShsm stores backup versions, dump copies, or migrated data sets.

DFSMSrmm. A DFSMS/MVS functional component or base element of OS/390, that manages removable media.

DFSMSrmm control data set. See control data set.

DFSMSrmm-managed volume. A tape volume that is defined to DFSMSrmm. Contrast with system-managed volume and DFSMShsm-managed volume.

disaster recovery. A procedure for copying and storing an installation's essential business data in a secure location, and for recovering that data in the event of a catastrophic problem. Compare with vital records.

discrete profile. A resource profile that provides RACF protection for a single resource.

DISTANT. A DFSMSrmm built-in storage location ID. See built-in storage location.

DSNB. Data set name block.

dual copy. A high availability function made possible by nonvolatile storage in some models of the IBM 3990 Storage Control. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical 3990 subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

dump class. A set of characteristics that describes how volume dumps are managed by DFSMShsm.

duplexing. The process of writing two sets of identical records in order to create a second copy of data.

dynamic cache management. A function that automatically determines which data sets will be cached based on the 3990 subsystem load, the characteristics of the data set, and the performance requirements defined by the storage administrator.

E

ECCST. Enhanced Capacity Cartridge System Tape.

EDM. External data manager.

EHPCT. Extended High Performance Cartridge Tape.

eject. The process used to remove a 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 dataserver, the volume is not moved, but the tape configuration database is updated to show the volume no longer resides in the manual tape library dataserver.

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.

EOV. End of volume.

EREP. Environmental Record Editing and Printing program.

expanded output. Expanded output occurs when you specify OUTPUT=FIELDS and EXPAND=YES. For those subcommands for which expanded output applies, your application program receives more variable data than for standard output.

expiration. The process by which data sets and volumes are identified as available for reuse. In DFSMSrmm, all volumes have an expiration date or retention period set for them either by vital record specification policy, by user-specified JCL when writing a data set to the volume, or by an installation default. When a volume reaches its expiration date or retention period, it becomes eligible for release.

expiration date. The date at which a file is no longer protected against automatic deletion by the system.

expiration processing. The process of inventory management that ensures expired volumes are released and carries out required release actions on those volumes.

export. The operation to remove one or more logical volumes from a virtual tape server library. First, the list of logical volumes to export must be written on an export list volume and then, the export operation itself must be initiated.

export list volume. A virtual tape server logical volume containing the list of logical volumes to export.

exported logical volume. A logical volume that has one through the export process and now resides on a tacked volume outside a virtual tape server library.

external label. A label attached to the outside of a tape cartridge that is to be stored in an IBM 3494 Tape Library Dataserver or IBM 3495 Tape Library Dataserver. The label might contain the DFSMSrmm rack number of the tape volume.

extract data set. A data set that you use to generate reports.

F

field format. Field format is where the output consists of Structured Field Introducers and variable data rather than output in line format.

filtering. The process of selecting data sets based on specified criteria. These criteria consist of fully or partially-qualified data set names or of certain data set characteristics.

FIPS. Federal Information Processing Standard.

FMID. Function modification identifier.

FRR. Functional recovery routines.

G

GDG. Generation data group.

GDS. Generation data set.

Archive

generation data group. A collection of data sets kept in chronological order. Each data set is a generation data set.

generation number. The number of a generation within a generation data group. A zero represents the most current generation of the group, a negative integer (-1) represents an older generation and, a positive integer (+1) represents a new generation that has not yet been cataloged.

generic profile. A resource profile that can provide RACF protection for zero or more resources. The resources protected by a generic profile have similar names and identical security requirements, though with RACFVARS, a generic profile can protect resources with dissimilar names, too. For example, a generic data set profile can protect one or more data sets.

giga (G). The information-industry meaning depends upon the context:
1. G = 1,073,741,824(2^30) for real and virtual storage
2. G = 1,000,000,000 for disk storage capacity (e.g., 4Gb fixed disk)
3. G = 1,000,000,000 for transmission rates.

GPR. General purpose register.

GRS. Global Resource Serialization.

guaranteed space. A storage class attribute indicating the space is to be preallocated when a data set is created. If you specify explicit volume serial numbers, SMS honors them. If space to satisfy the allocation is not available on the user-specified volumes, the allocation fails.

H

hardware configuration definition. An interactive interface in MVS that enables an installation to define hardware configurations from a single point of control.

HLQ. High-level qualifier.

HPCT. High Performance Cartridge Tape.

high capacity input station. A transfer station, used by the operator to add tape cartridges to the IBM 3494 Tape Library Dataserver or IBM 3495 Tape Library Dataserver, which is inside the enclosure area.

high capacity output station. A transfer station, used by the operator to remove tape cartridges from the automated tape library dataserver, which is inside the enclosure area.

home. See home location.

home location. For DFSMSrmm, the place where DFSMSrmm normally returns a volume when the volume is no longer retained by vital records processing.

I

IBM. International Business Machines.

ICETOOL. DFSORT's multipurpose data processing and reporting utility.

ICF. Integrated catalog facility.

ICF catalog. A catalog that is composed of a basic catalog structure (BCS) and its elated volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). See also basic catalog structure and VSAM volume data set.

ID. Identifier.

import. The operation to enter previously exported logical volumes residing on a stacked volume into a virtual tape server library. First, the list of logical volumes to import must be written on an import list volume and the stacked volumes must be entered, and then, the import operation itself must be initiated.

import list volume. A virtual tape server logical volume containing the list of logical volumes to import. This list can contain individual logical volumes to import and/or it can contain a list of stacked volumes in which all logical volumes on the stacked volume are imported.

Archived

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. 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/MVS that allows users and storage administrators access to the storage management functions.

Interactive System Productivity Facility (ISPF). An IBM licensed program used to develop, test, and run interactive, panel-driven dialogs.

in transit. A volume is in transit when it must be moved from one location to another and DFSMSrmm believes that the move has started, but has not yet received confirmation that the move is complete. For a volume moving from a system-managed library, the move starts when the volume is ejected.

internal label. The internal label for standard label tapes is recorded in the VOL1 header label, magnetically recorded on the tape media.

inventory management. The regular tasks that need to be performed to maintain the control data set. See also expiration processing, storage location management processing, and vital record processing.

IPCS. Interactive Problem Control System.

IPL. Initial program load.

ISMF. Interactive Storage Management Facility.

ISPF. Interactive System Productivity Facility.

ISO. International Organization of Standardization.

ITSO. International Technical Support Organization.

IVP. Installation verification procedure.

J

JCL. Job control language.

JES2. Job entry subsystem 2.

JES3. Job entry subsystem 3.

JFCB. Job file control block.

journal. A sequential data set that contains a chronological record of changes made to the DFSMSrmm control data set. You use the journal when you need to reconstruct the DFSMSrmm control data set.

K

keyword. A predefined word that is used as an identifier.

kilo (K). The information-industry meaning depends upon the context:
1. K = 1024($2^1$0) for real and virtual storage
2. K = 1000 for disk storage capacity (e.g., 4000 KB fixed disk)
3. K = 1000 for transmission ratesTerm2 definition.

KSDS. Key-sequenced data set.

L

LCS. Library Control System.

Library Control System. The Object Access Method component that controls optical and tape library operations and maintains configuration information.

Archived

Line format. Line format is where text and variable data are formatted into lines suitable for displaying at a terminal or printing on hardcopy output.

LOCAL. A DFSMSrmm built-in storage location ID. See built-in storage location.

location name. A name given to a place for removable media that DFSMSrmm manages. A location name can be the name of a system-managed library, a storage location name, or the location SHELF, identifying shelf space outside a system-managed library or storage locations.

logical volume. Logical volumes have a many-to-one association with physical tape media and are used indirectly by MVS applications. They reside in a Virtual Tape Server or on exported stacked volumes. Applications can access the data on these volumes only when they reside in a Virtual Tape Server which makes the data available via its tape volume cache or after the data has been copied to a physical volume through the use of special utilities.

low-on-scratch management. The process by which DFSMSrmm replenishes scratch volumes in a system-managed library when it detects that there are not enough available scratch volumes.

LPA. Link pack area.

LSR. Local shared resources.

M

management class. A collection of management attributes, defined by the storage administrator, used to control the release of allocated but unused space: to control the retention, migration, and backup of data sets: to control the retention and backup of aggregate groups, and to control the retention, backup, and class transition of objects. If assigned by ACS routine to system-managed tape volumes, can be used to identify a DFSMSrmm vital record specification.

management value. See vital record specification management value.

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 dataserver. DFSMSrmm can initiate this process.

manual mode. An operational mode where DFSMSrmm runs without recording volume usage or validating volumes. The DFSMSrmm TSO commands, ISPF dialog, and inventory management functions are all available in manual mode.

manual tape library. A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which can be mounted on those drives. See also tape library. Contrast with automated tape library.

master system. The MVS system where the master DFSMSrmm control data set resides.

master volume. A private volume that contains data that is available for write processing based on the DFSMSrmm EDGRMMxx parmlib MASTEROVERWRITE operand.

media format. The type of volume, recording format and techniques used to create the data on the volume.

media library. See removable media library.

media management system. A program that helps you manage removable media. DFSMSrmm is a media management system.

media name. An up to 8 character value that describes the shape or type of removable media stored in a storage location. Examples of media name are: SQUARE, ROUND, CARTRDGE, 3480.

media type. A value that specifies the volume's media type. Media type can be specified as: *, CST, ECCST, HPCT or EHPCT.

MEDIA 1. Cartridge System Tape.

MEDIA 2. Enhanced Cartridge System Tape.

MEDIA 3. High Performance Cartridge Tape.

MEDIA 4. Extended High Performance Cartridge Tape.

mega (M). The information-industry meaning depends upon the context:
1. M = 1,048,576(2æ0) for real and virtual storage
2. M = 1,000,000 for disk storage capacity (e.g., 4000 MB fixed disk)
3. M = 1,000,000 for transmission ratesTerm2 definition.

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled. See also migration level 1and 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 and migration level 2.

migration level 2. DFSMShsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also storage hierarchy. Contrast with primary storage and migration level 1.

MVS image. A single occurrence of the MVS/ESA operating system that has the ability to process work.

N

name vital record specification. A vital record specification used to define additional retention and movement policy information for data sets or volumes.

NEXTVRS. An RMM ADDVRS TSO subcommand operand. See Using NEXT.

NL. No label.

non-scratch volume. A volume that is not scratch, which means it has valid or unexpired data on it. Contrast with scratch.

NSL.. Non standard label.

O

OAM. Object Access Method.

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

OCDS. Offline control data set.

optical disk. A disk that uses laser technology for data storage and retrieval.

optical volume. Storage space on an optical disk, identified by a volume label. See also volume.

option line. See command line.

owner. In DFSMSrmm, a person or group of persons defined as a DFSMSrmm user owning volumes. An owner is defined to DFSMSrmm through an owner ID.

owner ID. In DFSMSrmm, an identifier for DFSMSrmm users who own volumes.

P

parallel running. During conversion, when you install DFSMSrmm concurrently with an existing media management system, it is called running in parallel.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

PDS. Partitioned data set.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with temporary data set.

PF. Program function key.

physical stacked volume. See stacked volume.

physical volume. Physical volumes have a one-to-one association with physical tape media and are used directly by MVS applications. They may reside in an automated tape library dataserver or be kept on shelf storage either at vault sites or within the data center where they 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.

pooling. The process of arranging shelf locations in the removable media library into logical groups.

pool ID. The identifier for a pool. You define pool IDs in parmlib member EDGRMMxx.

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also storage group.

PPT. Program properties table.

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 1and migration level 2.

private tape volume. A volume assigned to specific individuals or functions.

primary vital record specification. The first retention and movement policy that DFSMSrmm matches to a data set and volume used for disaster recovery and vital record purposes. See also vital record specification and secondary vital record specification.

profile. Data that describes the significant characteristics of a user, a group of users, or one or more resources. A profile contains a base segment, and optionally, a number of other segments.

protect mode. In protect mode, DFSMSrmm validates all volume requests.

pseudo-generation data group. A collection of data sets, using the same data set name pattern, to be managed like a generation data group. The ~ masking character is used in DFSMSrmm to identify the characters in the pattern that change with each generation.

PSW. Program status word.

PTF. Program temporary fix.

pull list. A list of scratch volumes to be pulled from the library for use.

PUT. Program update tape.

R

RACF. See 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.

RDS. Retention data set.

ready to scratch. This describes the condition where a volume is eligible for scratch processing while it resides in a storage location. Since no other release actions are required, the volume can be returned to scratch directly from the storage location.

record-only mode. The operating mode where DFSMSrmm records information about volumes as you use them, but does not validate or reject volumes.

recording format. For a tape volume, the format of the data on the tape; for example, 18 tracks, 36 tracks, 128 tracks or 256 tracks.

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a journal.

relative start generation. Relative generation zero is the latest generation of a tape; Relative generation -1 is the previous generation of that tape. Relative generation -2 is the generation before the previous one.

REMOTE. A DFSMSrmm built-in storage location ID. See built-in storage location.

removable media. See volume.

removable media library. The volumes that are available for immediate use, and the shelves where they could reside.

Resource Access Control Facility (RACF). An IBM licensed program that provides for access control by identifying and verifying the users to the system; authorizing access to protected resources; logging the detected unauthorized attempts to enter the system; and logging the detected accesses to protected resources.

Resource Group. A collection of structured fields that describe the attributes of a resource such as a volume.

Restructured Extended Executor (REXX) Language. A general-purpose, high-level programming language, particularly suitable for EXEC procedures or programs for personal computing.

retention date. Retention date can be the date that a data set or volume is retained by a vital record specification or the date of the inventory management run when the data set or volume is no longer retained by a vital record specification.

retention period. The time for which DFSMSrmm retains a volume or data set before considering it for release. You can retain a data set or volume as part of disaster recovery or vital records management. You set a retention period through a vital record specification that overrides a data set's expiration date.

retention type. The types of retention for which DFSMSrmm retains a volume or data set before considering it for release. The retention types for data sets are BYDAYSCYCLE, CYCLES, DAYS, EXTRADAYS, LASTREFERENCEDAYS, UNTILEXPIRED, and WHILECATALOG. The retention types for volumes are DAYS and CYCLE.

REXX. Restructured Extended Executor Language.

RMF. Resource Measurement Facility.

RMM. Removable Media Manager.

RMM complex (RMMplex). One or more MVS images that share a common DFSMSrmm control data set.

RMODE. Residence mode.

S

SAF. System Authorization Facility.

scratch. The status of a tape volume that is available for general use, because the data on it is incorrect or is no longer needed. You request a scratch volume when you omit the volume serial number on a request for a tape volume mount.

scratch pool. The collection of tape volumes from which requests for scratch tapes can be satisfied. Contrast with rack pool.

scratch processing. The process for returning a volume to scratch status once it is no longer in use and has no outstanding release actions pending.

scratch tape. See scratch volume.

scratch volume. A tape volume that contains expired data only. See scratch.

SDB. Structured database.

SDSF. Spool Display and Search Facility.

secondary space allocation. Amount of additional space requested by the user for a data set when primary space is full. Contrast with primary space allocation.

secondary vital record specification. The second retention and movement policy that DFSMSrmm matches to a data set and volume used for disaster recovery and vital records purposes. See also vital record specification and primary vital record specification.

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.

SMS. Storage management subsystem.

Archive

stacked volume. Stacked volumes have a one-to-one association with physical tape media and are used in a Virtual Tape Server to store logical volumes. Stacked volumes are not used by MVS applications but by the Virtual Tape Server and its associated utilities. They may be removed from a Virtual Tape Server to allow transportation of logical volumes to a vault or to another Virtual Tape Server.

standard access list. The portion of a resource profile that specifies the users and groups that may access the resource and the level of access granted to each.

standard label. An IBM standard tape label.

Standard output. Standard output is the amount of variable data displayed, printed or put into a REXX variable in response to a subcommand. When you specify OUTPUT=LINES or EXPAND=NO with OUTPUT=FIELDS, your application program receives standard output as opposed to expanded output.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD volumes and optical volumes treated as a single object storage hierarchy.

storage location. A location physically separate from the removable media library where volumes are stored for disaster recovery, backup, and vital records management.

(storage) location dominance. The priority used by DFSMSrmm to decide where to move a volume within the removable media library during vital record specification processing. It covers all the locations; SHELF, storage locations, and system-managed tape libraries.

storage location management processing. The process of inventory management that assigns a shelf location to volumes that have moved as a result of vital record processing. See also vital record processing.

stripe. In DFSMS/MVS, the portion of a striped data set that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

structured field. Output from the DFSMSrmm application programming interface consisting of a Structured Field Introducer and output data.

structured field introducer. An 8-byte entity that either introduces the beginning of a group of data or introduces output data that immediately follows the introducer.

subsystem. A special MVS task that provides services and functions to other MVS users. Requests for service are made to the subsystem through a standard MVS facility known as the subsystem interface (SSI). Standard MVS subsystems are the master subsystem and the job entry subsystems JES2 and JES3.

subsystem interface (SSI). The means by which system routines request services of the master subsystem, a job entry subsystem, or other subsystems defined to the subsystem interface.

SUL. IBM standard and user header or trailer label.

SVC. Supervisor call.

system-managed storage. Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications. See also DFSMS environment.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also tape library.

system-managed volume. A DASD, optical, or tape volume that belongs to a storage group. Contrast with DFSMShsm-managed volume and DFSMSrmm-managed volume.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T

tape configuration database (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 and automated tape library.

Tape Library Control System (TLCS). IBM program offering 5785-EAW. DFSMSrmm replaces TLCS.

Tape Library Dataserver. A hardware device that maintains the tape inventory associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, and storage of tapes. An automated tape library that supports system-managed storage of tape volumes. IBM's automated tape library dataservers include the IBM 3494 Tape Library Dataserver and the IBM 3495 Tape Library Dataserver.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also storage group.

tape subsystem. A magnetic tape subsystem consisting of a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume. A tape volume is the recording space on a single tape cartridge or reel. See also volume.

TCDB. Tape Configuration Database.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with permanent data set.

tera (T). The information-industry meaning depends upon the context:
1. T = 1,099,511,627,776(240) for real and virtual storage
2. T = 1,000,000,000,000 for disk storage capacity (e.g., 4 TB of DASD storage)
3. T = 1,000,000,000,000 for transmission rates.

TMC. Tape management catalog.

TRS. Tape retention system.

TSO. Time sharing option.

U

Until expired. Allows the use of vital record specification policies for managing retention in a location as long as the volume expiration date has not been reached.

use attribute. (1) The attribute assigned to a DASD volume that controls when the volume can be used to allocate new data sets; use attributes are public, private, and storage. (2) For system-managed tape volumes, use attributes are scratch and private.

user volume. A volume assigned to a user, that can contain any data and can be rewritten as many times as the user wishes until the volume expires.

using AND. A method for linking DFSMSrmm vital record specifications to create chains of vital record specifications. DFSMSrmm applies policies in chains using AND only when all the retention criteria are true.

using NEXT. A method for linking DFSMSrmm vital record specifications to create chains of vital record specifications. DFSMSrmm applies policies in chains using NEXT one vital record at a time.

V

virtual export. Mark 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 Magstar® 3494 Tape Library, combines the random access and high performance characteristics of DASD with outboard hierarchical storage management and virtual tape devices and tape volumes.

vital record group. A set of data sets with the same name that matches to the same DFSMSrmm vital record specification.

vital record processing. The process of inventory management that determines which data sets and volumes DFSMSrmm should retain and whether a volume needs to move. These volumes and data sets have been assigned a vital record specification.

vital record specification. Policies defined to manage the retention and movement of data sets and volumes used for disaster recovery and vital records purposes.

vital record specification management value. A one-to-eight character name defined by your installation and used to assign management and retention values to tape data sets. The vital record management value can be any value you chose to create a match between a vital record specification and data sets and volumes in your installation. By matching the vital record specifications to the data set or volumes, DFSMSrmm applies the retention and movement policies you define in the vital record specifications. During inventory management VRSEL processing, DFSMSrmm selects the correct, best matching vital record specification for a tape data set or volume.

vital records. A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement. Compare with disaster recovery.

VMF. Volume master file.

VMS. Volume management system.

VNL. Volume not in library.

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). An identification number in a volume label that is assigned when a volume is prepared for use on the system. For standard label volumes, the volume serial number is the VOL1 label of the volume. For no label volumes, the volume serial number is the name the user assigns to the volume. In DFSMSrmm, volume serial numbers do not have to match rack numbers.

VPDD. Vault pattern description data set.

VRS. Vital record specification.

VTS. Virtual Tape Server.

VSAM volume data set (VVDS). A data set that describes the characteristics of VSAM and system-managed data sets residing on a given DASD volume; part of an integrated catalog facility catalog. See also basic catalog structure and integrated catalog facility catalog.

W

warning mode. The operating mode in which DFSMSrmm validates volumes as you use them, but issues warning messages when it discovers errors instead of rejecting volumes.

write-to-operator. 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. See write-to-operator.

Archived

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 698.

- *Converting to Removable Media Manager: A Practical Guide*, SG24-4998
- *DFSMSrmm Primer*, SG24-5983
- *Merging Systems into a Sysplex*, SG24-6818

Other resources

These publications are also relevant as further information sources:

- *z/OS DFSMSrmm Implementation and Customization Guide*, SC26-7405
- *z/OS DFSMSrmm Guide and Reference*, SC26-7404
- *z/OS DFSMSrmm Reporting*, SC26-7406
- *z/OS SecureWay Security Server RACF Security Administrator’s Guide*, SA22-7683
- *z/OS MVS System Messages, Vol. 5 (EDG-GFS)*, SA22-7635
- *z/OS MVS System Management Facilities (SMF)*, SA22-7630
- *z/OS MVS Initialization and Tuning Reference*, SA22-7592
- *z/OS TSO/E Customization*, SA22-7783

Referenced Web sites

These Web sites are also relevant as further information sources:

- z/OS information wizardry:

  http://www.ibm.com/servers/eserver/zseries/zos/wizards/

- OS/390 information wizardry:

  http://www.ibm.com/servers/s390/os390/wizards/

© Copyright IBM Corp. 2004. All rights reserved.

How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

ibm.com/redbooks

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Archived

Index

Numerics
3420 143
3480 cartridge system tape 143, 484
3490 cartridge system tape 143, 484
3494 Automated Tape Library Dataserver 479
3495 Automated Tape Library Dataserver 479
3590 High Performance Tape Subsystem 143

A
ABEND 407
access method services
  REPRO 452
ACF2 DFSMSrmm definitions 674
ACS processing
  pooling decisions 165
ACS routine 223
ACS routines 140, 159, 228, 406
  EDGRMMxx OPTION subparameters 138
  EDGUX100 user exit 139
  how it works 140
  MSPOOL variable 141
  PREACS processing 142
  pre-ACS processing 141
  PREACS subparameter 139
  scratch pooling 140
  SMSACS subparameter 139
  VRSs in MC routine 165
addition of a VRS name 300
additional functions 388
allocation 9
alter volume ownership 300
analyzing the record layouts 209
ANDVRS 234
ANDVRS operand 416
API
  See Application Programming Interface
Application Programming Interface 14
ARCTVEXT exit 61, 135, 217
assigning media type 278
ATL
  See automated tape library
audit records 79
authorization and security 322
automated tape library

B
Basic Tape Library Support 143, 479
  LIBRARY command 480
  scratch pool 479
bin assignment 352
bin number 19
BLP and NL processing 216
boxes 202
BronzePlex
  considerations for RMMplex 583
BTLS
  See Basic Tape Library Support

C
cabinets 202
CA-Disk SYSPARM 321
cartridge change use attributes exit 61, 133
cartridge eject exit 61, 133
cartridge entry exit 61, 133
catalog interface 12
catalog synchronization 340
  authorization required 330
  disabling 337
  EDGUTIL utility 329
  enabling 331
  enabling JCL 339
  journal considerations 339
  LISTCONTROL subcommand 338, 341
  message data set 338
CA-Top Secret definitions 670
CBRUXCUA exit 12, 61, 134
CBRUXEJC exit 12, 61, 134
CBRUXENT exit 12, 61, 134, 169–170
CBRUXVNL exit 12, 61, 134
CDS
  See control data set
CDS merge 572
  LOCDEF command 570
  PARMLIB options 570
  REJECT command 571
  VLPOOL command 571

© Copyright IBM Corp. 2004. All rights reserved.

CDS split 576
LOCDEF command 570
PARMLIB options 570
REJECT command 571
VLPOOL command 571
CDS-driven tape erasing and labeling 389
centralize control of retention and movement 385
change management 48
changing media name 282
checklist 607
cleaning up 383
collecting information 208
command running 306
comparing scratch lists 354
concurrent copy 452
confirm movements 124
control data set 2
action record 563
backup 438, 452
backup and recovery 4
bin record 568
contents 563
control record 563
create 91
data set record 564
extended format 95
forward recover 128
initialize 94
mend 475
merge 562
overview 7
owner record 564
product record 565
rack record 566
restore 127
space allocation 304
split 562
verify 131, 305, 438, 474
volume record 569
VRS record 564
control data set and journal backup 323
control information display 110
conversion
adding empty bins 306
adding VRS 307
analyzing current environment 55
completing 56
conversion plan 52
cutover 56
data unavailable 192
extracting data 55
full exploitation 57
learning 54
macros 207
measuring success 57
parallel running and validation 56
planning 48
process 42, 49
process flow 52
programs and exits 206–207
sample exits 207
sample schedule 53
stages 44, 51
starting DFSMSrmm 54
conversion checklist 607
create CDS 302
create extract data set 323
cross-reference 193
customizing DFSMSrmm
enable TSO HELP 109
cutover to production 361

D

data set
cycle 402
data set data fields 192
data set list 349
data set mask 238
data set name mask 384
data set overwrite 215
data set retention 32, 225
data set VRS 402
database synchronization 354
DB2 logging and backup 377
default media name 283
default retention days 212
default retention period 72, 317
define pools 75
defining locations 276
delete non-existent volumes 387
deletion policies 203
DFRMMLOC statement 238
DFSMS ACS support 419
DFSMSdfp 10
DFSMSdss 11
DFSMSdss CDS backup 129
DFSMSdss concurrent copy 663

Archived

DFSMShsm 11
consistency checking 478
DFSMShsm and DFSMSrmm interface 135
DFSMShsm interface 217
DFSMShsm tapes 418
TVEXTPURGE 418
DFSMSrmm
interfaces 8
reports 485
retrieving information 485
structure 7
DFSMSrmm CDS control record 304
DFSMSrmm education 136
DFSMSrmm PARMLIB options 209
DFSMSrmm policies 225
DFSMSrmm reporting
audit reports 526
creating resource lists 487
data set display 511
data set lists 495
DFSMSrmm options display 521
display information 505
EDGRPTD utility 443
EDGRRPTE exec 443
fast path commands 487
inventory management reports 526
inventory reports 529
ISPF dialog 486
movement reports 529
program products lists 493
rack and bin display 515
rack and bin lists 487
report generator 540
resource lists 486
resources information 486
RMM TSO subcommands 486
scratch lists reports 529
security reports 525
sorting resources lists 500
supplied reports 529
using DFSORT 539
using line operators on lists 504
utilities 525
virtual record specifications display 518
virtual record specifications lists 497
volume display 506
volume lists 489
DFSMSrmm supplied reports
creating 530
REPORT01 536
REPORT02 537
REPORT03 537
REPORT04 537
REPORT05 537
REPORT06 537
REPORT07 537
REPORT08 537
REPORT09 538
REPORT10 538
REPORT11 538
REPORT12 538
REPORT13 538
REPORT14 539
REPORT15 539
DFSORT symbols 539
disable catalog syncronization 337
disaster recovery 232
DISPLAY SMS command 153
DISPMSGID 70
disposition processing 464
DSNZPARM 378
duplicate data set names 589
duplicate records 304
duplicate RMF definitions 230
duplicate volsers 586, 589, 593
identifying 588

E

EDG019VM SAMPLIB member 135
EDGAUD utility 525
EDGBKUP utility 127, 212
EDGCDREC data set name information 198
EDGCDREC record layout 615
EDGCDYNM extract program 222, 230
EDGCDYNM input files 244
EDGCDYNM input parameters 245
EDGCDYNM OPTIONS statement 239
EDGCDYNM output files 244
EDGCDYNM parameters
DEFOWNER statement 252
DFRMMLOC statement 250
OPTIONS statement 245
SKIPVOL statement 252
TLMSLOCN statement 249
VOLMEDIA statement 251
VRSMGMTP statement 248
VRSNAMEP statement 247

VRSOWNER statement 253
EDGCDYNM program 243
execution 253
limitations 255
messages 255
return codes 255
EDGCEBIN program
execution parameter 296
input files 296, 299
output files 296, 299
processing 295
return codes 297, 299
EDGCEREC empty rack and bin information 200
EDGCEREC record layout 620
EDGCKREC record layout 623
EDGCKREC vital record Information 199
EDGCLREC record layout 628
EDGCLREC volume information 193
EDGCMM01 SAMPLIB member 242
EDGCNVT 290
EDGCNVT program
output files 284
preparation 264
return codes 286
EDGCNVT-00090 291
EDGCNXIT input 301
EDGCNXIT output 301
EDGCNXIT return codes 301
EDGCNXIT user exit 300
EDGCOREC owner record information 201
EDGCOREC record layout 643
EDGCVRSX SAMPLIB member 228
EDGDFHSM program interface 11
EDGHSKP program
BACKUP parameter 327
CATSYNCH parameter 327
DATE parameter 328
DATEFORM parameter 328
DSSOPT file 326
DSTORE parameter 328
EXPROC parameter 328
RPTEXT parameter 328
VERIFY parameter 328
VRSEL parameter 328
EDGHSKP utility 122, 212, 323, 434, 526
VERIFY operand 421
EDGINERS 483
EDGINERS utility 218, 463
EDGJLOAD SAMPLIB member 302
EDGJVERR SAMPLIB member 304
EDGLABEL procedure 126
EDGMSGEX exit 10
EDGRMMxx
MNTMSG command 75
OPTION command 69
REJECT command 74
SECCLS command 75
VLPOOL command 75
EDGRMMxx member 588, 594
EDGRMMxx PARMLIB member 68, 311
ACS processing 162
EDGRPTD report utility 368
EDGRPTD utility 443, 527
EDGRRPTE exec 443, 529
EDGRRPTE REXX 533
EDGTVEXT program interface 11
EDGTVEXT programming interface 61
EDGUTIL program 588, 590–592
EDGUTIL utility 213, 304, 329
EDGUX100 exit 132, 169, 219, 223, 419
EDGUX100 user exit 139
EDGUX200 exit 133, 169
EDRPTD report utility 329
EJECT command 371
enabling 340
enabling catalog synchronization 331
entry 24
exclusion list 80
exits
CBRUXCUA 12
CBRUXEJC 12
CBRUXENT 12
CBRUXVNL 12
EDGMSGEX 10
EDGUX100 419
IGDACSXT 420
IGXMSGEX 10
testing 133
expiration date 364
expiration date decisions 157
expiration processing 172, 323
expiration processing housekeeping function 668
EXPIRYDATEIGNORE 234
extract data set 122
extraction
preparation 243
EXTRADAYS 225

F

fall back plan 382
fast path commands
  CONTROL SYSTEM 505
  DATASET DISPLAY 505
  DATASET SEARCH 487
  OPTIONS SORT 500
  OWNER DISPLAY 505
  PRODUCT DISPLAY 506
  PRODUCT SEARCH 487
  RACK DISPLAY 505
  RACK SEARCH 487
  VOLUME DISPLAY 505
  VOLUME SEARCH 487
  VRS DISPLAY 505
  VRS SEARCH 487
fast tape positioning 13
final production
  cleaning up 383
final testing
  DB2 logging and backup 377
  RACF tape protection 378
  SMS ACS support 377
foreign and product tape management 388
foreign tapes
  define 467
foreign volumes 25
functions comparison 185

G

GDG
  See generation data group
generation data group 405
global confirmation 444
global resource serialization 80
GoldPlex
  considerations for RMMplex 584
GRS
  See global resource serialization
GRSRNLxx PARMLIB member 80

H

high speed locate 13
home location 19, 32, 36
housekeeping 3, 401, 434
HSC initialize cartridge utility 371
HSC scratch conversion 372
HSC SLUADMIN utility 344
HSC/MVS 371, 482
HSM DELVOL command 135

I

ICETOOL 539
IDCAMS REPRO 571
identify system-managed volumes 265
IEFSSNxx PARMLIB member 62, 310
IFAPRDxx PARMLIB member 67
IFG019VM tape exit 135
IGDACSXT exit 420
IGXMSGEX exit 10, 61
IKJTSOxx PARMLIB member 65
implementation 60
init 24
installation verification program 105
installation wizard 60
installation-critical data 232
installation-wide exits 61, 214, 321
interface to tape libraries 368
interfaces
  batch TSO 343
  BTLS 12, 342
  BTLS interface 241
  short-on-scratch processing 12, 241, 342
  SMS 342
  SMS interface 241
  STK 343
  TSO and ISPF 241, 343
  TSO batch 242
inventory by location 350
inventory management 434, 482
  activities 3
  allocating data sets 122
  BACKUP 123
  CDS and journal backup 123
  CDS backup 438
  CDS verify 438
  daily applications 436
  DSTORE 123
  expiration processing 123, 439
  EXPROC 123, 172
  extract data set processing 123
  housekeeping 439
  monthly applications 436
  open data sets 444
  permanent I/O errors 444
  preparation 437

Archive

report extract 440
RPTEXT 123
running 123
scheduling tasks 435
storage location management 123, 439
trial run 440
vital record processing 123
vital records processing 439
VRSEL 123
weekly applications 436
inventory management and reports 212
inventory management reports
ACTIVITY file 526
extract data set 526
vital record specification report 526
IPL considerations 366
ISPF dialog 14

J

job mask 238
job name creation 354
journal
activate 349
create 97
disable 321, 347
EDGJNLAL SAMPLIB member 98
forward recovery 128
JOURNALFULL parameter 346
journal data set
overview 7
journal percentage full threshold 666
journal threshold 316

L

LASTREFERENCEDAYS 407
library management 21, 37, 41
library name 17
limiting business risk 49
LISTCONTROL subcommand 338
LISTDATASET output 330
loan location 15
local SYSTEM enqueue 80
logical file sequence number 353

M

management class 223, 405–406
management value 402, 405
manual tape library
master volume 24
Memorex tape libraries 375
merging and splitting your CDS 389
messages
CBR3006I 150
CBR3010I 379
DXT002 255
DXT010 255
DXT011 256
DXT012 256
DXT013 256
DXT014 256
DXT020 256
DXT022 257
DXT024 257
DXT030 257
DXT031 258
DXT032 258
DXT040 258
DXT092 259
DXT093 259
DXT094 259
DXT095 259
DXT096 259
DXT097 260
DXT098 260
EDG0103D 101, 345
EDG0105I 101, 347
EDG0122I 347
EDG0181I 104
EDG0204I 347
EDG1113I 103
EDG1114I 103
EDG1118I 103
EDG2103D 347
EDG2104E 347
EDG2107E 346
EDG4027I 377
EDG4050I 377
EDG4054I 465
EDG8121D 134
EDG8122D 134
EDG8123D 134
EDG8124I 134
EDGCEBIN-00000 297
EDGCEBIN-00005 297
EDGCEBIN-00010 297
EDGCNVT-00000 286

Archived

EDGCNVT-00005 287
EDGCNVT-00010 287
EDGCNVT-00015 287
EDGCNVT-00020 288
EDGCNVT-00025 288
EDGCNVT-00030 288
EDGCNVT-00035 288
EDGCNVT-00040 288
EDGCNVT-00045 289
EDGCNVT-00050 289
EDGCNVT-00055 289
EDGCNVT-00060 289
EDGCNVT-00065 289
EDGCNVT-00070 290
EDGCNVT-00075 290
EDGCNVT-00080 290
EDGCNVT-00085 290
EDGCNVT-00095 291
EDGCNVT-00100 291
EDGCNVT-00105 291
EDGCNVT-00110 292
EDGCNVT-00115 292
EDGCNVT-00120 292
EDGCNVT-00125 292
EDGCNVT-00130 293
EDGCNVT-00135 293
EDGCNVT-00140 293
EDGCNVT-00145 293
EDGCNVT-00150 294
EDGCNVT-00155 294
IDC3314I 304
IDG008I 157, 159
IEC507D 365
IEF352I 104
IGD008I 153
mount and fetch 75
migration wizard 61
MIM 84
MIMQNAME 84
MODIFY DFRMM command 163
MOVEBY 231
movement 36
movement confirmation 239
movement control 388
movement policies 31, 238
DELAY 415
LOCATION 414
PRIORITY 415
STORENUMBER 414
MSPOOL variable 141
MTL
See manual tape library
multi-data set multi-volume retention 231
multiple scratch pool support 219
multi-volume chain 353
multi-volume set 231, 353
MOVEBY 401
RETAINBY 394
MVS MODIFY command 101

N

name VRS 409
never-scratch 225
NEXTVRS 225
NEXTVRS operand 416
NL scratch tapes 135
non-GDG processing 231
non-IBM libraries 481
non-system managed volumes 140
non-system-managed tape libraries 143
non-system-managed tape library 19
notification 389

O

OAM
See Object Access Method
OAM exit processing 364
Object Access Method 12, 153
OCE
See open/close/end of volume
OPEN 407
open/close/end of volume 9
operating DFSMSrmm
initialize and erase volumes 463
processing labels 464
quiesce 102
refresh 102
remove 103
restart 101
shut down 104
start 101
stop 102
operation mode
protect 358
record-only 358
record-only mode 69
warning 358

Archive

warning mode 314
operator procedures 136
OPTION command
BACKUPPROC 69, 315
DATEFORM 69–70, 315
DISPMSGID 70
DSNAME 70, 315
JOURNALFULL 316
JRNLNAME 70, 315
MASTEROVERWRITE 216
MAXHOLD 316
MAXRETPD 71, 316
MOVEBY 317, 362
OPMODE 69, 314, 361
PREACS 71, 139, 317
RETAINBY 317, 362
RETPD 72, 317
REUSEBIN 72, 317
SCRATCHPROC 318, 320
SMFAUD 79
SMFSEC 79
SMSACS 139
SMSTAPE 72, 318, 362
SYSID 470
TPRACF 73, 220, 319, 363
TVEXTPURGE 217, 319, 363
UNCATALOG 73, 320, 363
VRSCHANGE 320, 364
VRSEL 74, 320
VRSJOBNAME 74, 320
Owner record 590

P

parallel running
authorization and security 322
check interfaces 342
comparing scratch lists 354
data set list report 349
differences 351
EDGHSKP first run 346
EDGHSKP program 323
inventory by location 350
inventory management 323, 345
JOURNALFULL parameter 346
multi-volume reports 351
preparation 310
processing 344
scratch volume lists 350
start DFSMSrmm 345
starting procedure 321
storage location bins 352
tape and inventory management 345
validating DFSMSrmm operation 349
vital record selection 346
PARMLIB members
DEVSUPxx 169
PARMLIB options
OPTION MOVEBY 401
OPTION RETAINBY 394
OPTION TVEXTPURGE 418
OPTION VRSEL(NEW) 409
OPTION VRSEL(OLD) 409
REJECT 39
SECCLS 525
VLPOOL 466, 469
VRSJOBNAME 398
PARMLIB options display 110
partition an SMS library 169
partitioning a library 171
pending release 24, 214
percentage full threshold 666
physical file sequence number 353
PlatinumPlex
considerations for RMMPlex 584
policy management
housekeeping 401
matching data sets 405
matching volumes 408
overview 391
retention types 410, 412–413
VRS modification 420
VRS parameters 409
VRS types 401
pooling types 469
pools
define 469
PREACS processing 142
pre-ACS processing 141
pre-ACS processing benefits 169
pre-ACS support 420
primary VRS 225
production cutover
activate exits 365
catalog syncronization 368
checklist 376
commitment to DFSMSrmm 380
final testing 377

inventory management 367
modifying CA-Disk 366
OPTION command 361
PARMLIB options 358
starting DFSMSrmm 375
tape library 369
VLPOOL command 364
VRS cleanup 384
protect tape volumes 445

R

RACF 11
assigning a user ID 648
authorize ABARS to DFSMSrmm resources 665
authorize DFSMShsm to DFSMSrmm resources 664
automatic tape security support 447
control access 365
defining ABARS procedure names 665
defining DFSMShsm procedure names 664
DFSMShsm and DFSMSrmm considerations 449
DFSMSrmm access list 6
DFSMSrmm resource profiles 658
DFSMSrmm user ID 89
discrete profiles 93, 99, 655
discrete TAPEVOL profile 447
EDGBKUP procedure 667
EDGLABEL procedure 669
EDGXPROC procedure 668
enhanced generic naming considerations 655
FACILITY class 91, 217, 220
identify started procedure 90
implementation 89
multi-system considerations 664–665
parallel running 322
protecting DFSMSdss resources 663
refresh instorage profiles 450
RESET facility 91
started task user ID 648
STGADMIN.EDG.RESET.SSI profile 91
tape profiles 73, 319
tape support 76
TAPEDSN class 220, 445
TAPEVOL class 220, 445
TAPEVOL profiles 447, 449
TPRACF operand 447
TVTOC 448
VLPOOL command 448
RACF tape protection 377–378
RACF tape security support 447
rack number 16
rack pool 75, 468
record layout description 614
Redbooks Web site 698
Contact us xxxv
reduce VRS policies 383
REJECT command 39
ANYUSE 74, 169
RELEASE 234
release options
EXPIRYDATEIGNORE 415
SCRATCHIMMEDIATE 415
removable media library 15
report generator 213
implementation 543
installation library 541
product library 541
report definition 540, 547
report type 540, 544
reporting tool 541, 545
user library 541
using 547
reports
creation 125
DFSMSrmm-supplied reports 125
EDGAUD 126
EDGAUD utility 125
EDGRPTD 125
EDGRPTD utility 125, 329
multifile multi-volume 351
report generator 126
REPORT06 350
REPORT11 351
searches and lists 4
reserve conversion list 81
resource 486
RETAINBY 231
retention and movement 221
retention limits
COUNT 413
DELETEDATE 414
retention method 203
retention methods 188
retention period 71
retention policies 31

retention types
BYDAYSCYCLE 411
CYCLES 410
DAYS 411
EXTRADAYS 412
LASTREFERENCEDAYS 411
UNTILEXPIRED 411
WHILECATALOG 411
RMF TLMS016 report 243
RMM ADDPRODUCT subcommand 466
RMM ADDRACK subcommand 466
RMM ADDVOLUME subcommand 179
RMM ADDVRS subcommand 405
RMM CHANGEVOLUME subcommand
CONTAINER 179
RMM LISTCONTROL subcommand 524
RMM LISTDATASET subcommand 514
RMM LISTRACK subcommand 517
RMM LISTVOLUME subcommand 510
RMM LISTVRS subcommand 520
RMM SEARCHDATASET subcommand 497
RMM SEARCHPRODUCT subcommand 495
RMM SEARCHRACK subcommand 487
RMM SEARCHVOLUME subcommand 482, 492
RMM SEARCHVRS subcommand 500
RMMCLIST data set 370
RMMPLEX 590
RMMplex
Action records 588
Bin numbers record 593
CDS record descriptions 586
CDSID value 588
considerations for BronzePlex 583
considerations for GoldPlex 584
considerations for PlatinumPlex 584
Control record 588
Data set record 589
duplicate data set names 589
duplicate volsers 586, 589, 593
EDGRMMxx member 588, 594
EDGUTIL program 588, 590–592
handling duplicate CDs records 587
LOCDEF keyword considerations 595
merge job 586
Move records 588
OPTIONS keyword considerations 594
Parmlib options 594
Product record 591
Rack numbers record 592
REJECT keyword considerations 595
toleration service 585
VLPOOL keyword considerations 595
VRS record 589
RMMSCR procedure 12
robot interface 377–378
router exit 61

S

sample extraction programs 242
SAMPLIB members
EDGCVRSX 432
EDGJACTP 421
EDGJBKP1 438
EDGJBKP2 438
EDGJCMOV 444
EDGJDHKP 441
EDGJEJC 442
EDGJEXP 441
EDGJLOPC 434
EDGJMOVE 443
EDGJRPT 531
EDGJSCRL 442
EDGJSRDS 432
EDGJVFY 438
EDGJWHKP 441
EDGLABEL 463
EDGPHKPA 437
scratch categories 169
scratch list report 213
scratch pool 75
scratch pooling 469, 471
scratch volume 24
scratch volume lists 350
SCRATCHIMMEDIATE 234
secondary VRS 225
secure volumes 525
security 6, 220, 322
security classes 75
security classification 6
security policies 647
security profiles 363
security records 79
SET PROD operator command 67
SETSMS command 156
SETSSI command 63
shelf location 16
SLSUX06 exit 484

Archived

SLUADMIN utility 371, 483
SLUCONDB program 483
SLUCONDB utility 372
SLUDRRMM routine 483
SMA ACS support
  &MSPARM 166
SMF security records 75
SMFPRMxx PARMLIB member 79
SMS ACS processing
  VRS definitions 164
SMS ACS routines 138
SMS ACS suport
  tailoring ACS routines 159
SMS ACS support 35
  &ACSENVIR variable 144, 166
  &MSDEST variable 166
  &MSPOLICY variable 166
  &MSPOOL variable 166
  EDGUX100 exit 139, 166–167
  expiration date decisions 157
  how it works 140
  IGDACSXT exit 166
  implementing 149
  management class define 157
  PREACS 139
  pre-ACS interface 167
  read only variables 143
  scratch pooling 139–140, 165
  SMSACS 139
  tape library define 150
  tape storage group define 153
SMS interface
SMS policy 143
software interfaces 9
software maintenance levels 48
software products
  define 466
special dates 223
specifying and translating owner information 275
specifying security levels 274
SSI
  see Subsystem Interface
stacked volumes 175
stacked volumes support 174
  container 176
  non-VTS 179
starting procedure 85, 321
STGADMIN.EDG.RESET.SSI FACILITY 91, 322
sticky labels 464
STK tape libraries 370, 482
STK VSM library 374
storage location 15
  built-in 19
  installation-defined 19
storage location bins 352
storage location management 21, 37, 41, 323, 482
subsystem interface 7, 345
subsystem name 62
system-managed tape
  exit testing 134
  library name 17
  OAM exits 133
  SMSTAPE operand 72, 318
system-managed tape libraries 38, 368
  partitioning 171–172
system-managed tape library 17
system-managed volumes
  IF STORLOC 265
  IF VOLRANGE 265
  LOCDEF 264

T

tape configuration database 17, 318
tape data sets
  record processing 25
tape drive cartridge loader 161
tape initialization 218
tape library support 370
tape pool management 219
tape replacement management 389
tape security support 447
tape usage 135
TAPEDSN class 378
TAPEDSN RACF class 445
TAPEVOL class 378
TAPEVOL RACF class 445
target destination 36
TCDB 364, 368
  See tape configuration database
terminology comparison 184
timing of events in your installation 48
Tivoli OPC 367, 434
  applications 435
  batch loader 434
  calendars 435
  ejecting volumes 442
  erasing and labeling volumes 441

Archived

Event-Triggered Tracking 435
move confirmation 443
operation 435
producing reports 443
scratch processing 441
scratch reporting 442
special resources 435
workstations 435
Tivoli Storage Manager 11
Tivoli Workload Scheduler 323
TLMS batch facilities 212
TLMS batch inventory update 209
TLMS data extraction 243
TLMS data set name types 221
TLMS exits 208
TLMS interfaces 208
TLMS locations 229
TLMS O/C/EOV interface 202
TLMS PARMLIB options 208–209
TLMS policies 221
TLMS reports 208
TLMS retention methods 188
TLMS shelf management 202
TLMS storage locations 232
TLMS structure 202
TLMS VMF 243
TLMSDTAB table 212, 221
TLMSLOCN RMMLOC keyword 238
TLMSLOCN statements 229
TLMSOPTS program 210
TLMSRACF interface 220
TLMSRTAB table 219
TLMSVCVS 243
TMSRPT2 utility 350
TPRACF operand 447
translating location names 272
TSM
See Tivoli Storage Manager
TSO commands 14
TSO RMM subcommand
ADDBIN 118
ADDRACK 115
ADDVOLUME 116
ADDVRS 119
CHANGEVOLUME 124
LISTOWNER 113
LISTVRS 119
SEARCHBIN 118
SEARCHRACK 115
SEARCHVOLUME 116
TSO RMM subcommands
ADDOWNER 113
authorization 65
LISTCONTROL 110
LISTCONTROL OPTION 348
TVEXTPURGE 418
TVEXTPURGE operand 135

U

UNCATALOG option 330
uncatalog processing 73
uninstalling Control-T 383
UNTILEXPIRED 226
user address space 7
user volume 24
using LOCDEF 276
using OWNER 275
using STORLOC 272
using VMEDIA 282
using VOLPRE 278
using VOLRANGE with LOCATION 278
using VOLRANGE with NORACK 281
using VOLRANGE with PREFIX 279
using VOLRANGE with SGNAME 280
using VOLRANGE with TYPE 281
utilities 212
authorization 65
EDGAUD 105
EDGHSKP 3, 104
EDGINERS 105, 126
EDGRPTD 105, 125, 368
EDGUTIL 94, 105, 131
setup 104
utlities
EDGAUD 126
UXTABLE 432

V

validation 26
vault 15
vendor produt removal 383
VERIFY(SMSTAPE) 475
VERIFY(VOLCAT) 475
Virtual Tape Server 174
export processing 176
import processing 177
virtual tape solutions 174

Archived

vital record processing 323
OPTION VRSEL 403
vital record specification
defining 164
delete 420
management value 402
primary VRS 402
recommendations 430
secondary VRS 402
trial run 421
vital record specifications 31
VLPOOL command 469
EXPDTCHECK 364
NAME 220
RACF 76
SYSID 219, 470
TYPE 75
VLPOOL RACF command 363
VLPOOL statement 219
VNL exit 319
VOLCAT 17
VOLSER mask 238
volume
automatic init 126
definition 133
entering the library 133
entry 24
erase 441
expiration date 352
init 24
initialize 441
life cycle 24
logical 23
management 23
master 24
pending release 12, 24
physical 23
scratch 24
shelf resident 19
stacked 23
status 24
user 24
volume data fields 190
volume not in library exit 61, 133
volume retention 34, 226
volume serial number 408
volume VRS 408
VRS
See vital record specification
See vital record specifications
VRS chain 409, 415
ANDVRS 415
NEXTVRS 415
VRS management class 34
VRS management value 34, 215, 223, 238, 406
VRS parameters 409
VRS processing 320
VRS subchain 416
VRS type
name 409
VRS types 401
data set 402
volume 408
VRSEL operand 403
VRSEL processing 330
VRSEL(NEW) 404
VSAM Extended Addressability 95
VTCS commands 375
VTS
See Virtual Tape Server
VTS stacked volume support 175

W

WHILECATALOG 406

Archived

Archived

Converting to DFSMSrmm from TLMS

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

REDBOOKS

Archived

Archived

Converting to DFSMSrmm from TLMS

Redbooks

Learn about the process and tools for converting to DFSMSrmm

Prepare a successful conversion plan

Run DFSMSrmm in paralel while converting

DFSMSrmm is the IBM tape management system for OS/390 and z/OS platforms. As part of DFSMS, DFSMSrmm is completely integrated into the IBM storage management strategy. This allows easier installation and maintenance, as well as standard interfaces with other systems components, such as DFSMSdfp and DFSMShsm.

DFSMSrmm provides a simple and flexible tape management environment, with support for all tape technologies, including IBM automated tape libraries, manual tapes, and other tape libraries.

This IBM Redbook is written for people who are planning to convert from TLMS to DFSMSrmm. We have designed this book to help you with all aspects of the conversion, from the early planning stage through implementation and customization of DFSMSrmm into your production system.

We provide details on the differences between DFSMSrmm and TLMS and compare the terminology, data, and functions. We explain how to use the IBM-supplied sample conversion programs and get the converted data validated and ready for use in a production environment. Working samples that are ready for use are included.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6242-01 ISBN 0738498920